

Quality Estimation of Workers in Collaborative Crowdsourcing Using Group Testing

Prakhar Ojha

Indian Institute of Science, Bangalore
prakhar.ojha@csa.iisc.ernet.in

Partha Talukdar

Indian Institute of Science, Bangalore
ppt@cds.iisc.ac.in

Abstract

Crowdsourcing is increasingly being used to solve complex tasks that require contributions from groups of individuals. In this paper, we consider the problem of distinguishing workers from idlers (who do not contribute positively) in group-based tasks. We consider a group as our smallest observable unit that can be evaluated and assume no knowledge of individual participant's contribution. We propose the use of group testing based methods for estimating quality of an individual, based on the performance of teams they have been part of. We further extend these algorithms to identify subsets of workers and give theoretical analysis on size of these subsets. We account for several real-world constraints in our model and present empirical support to our theoretical guarantees by an array of simulation experiments.

Introduction

Crowdsourcing has evolved from solving simpler tasks, like image-classification, to more complex tasks such as document editing, language translation, product designing etc (Rahman et al. 2015b). Unlike *micro-tasks* performed by a single worker, these complex tasks require a group of workers and greater resources. Recent research has highlighted the importance of *group-effort* to accomplish complex tasks, since individual worker's efforts are insufficient (Kittur et al. 2011). For instance, the task of collaborative document-writing (Figure 1) broadly involves planning, composing prose, writing coherent paragraphs etc. It is difficult to get an individual to write the entire document with low monetary costs. Such tasks fundamentally require coming together of multiple skills. The requester can either decompose the complex task into its constituent micro-tasks and assign them to individual workers (Little et al. 2009) or pose less decomposable tasks in entirety (Chamberlain 2014). Either way, groups of workers are involved who collaborate together and iteratively work over each others' contributions for completing the task.

The joint efforts of participants in a collaborative task are often so fundamentally interlinked that it is hard to separate out contribution of an individual. For example, in collaborative writing, participants repeatedly pursue different tasks like deciding the scope of article, collecting relevant facts,

Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

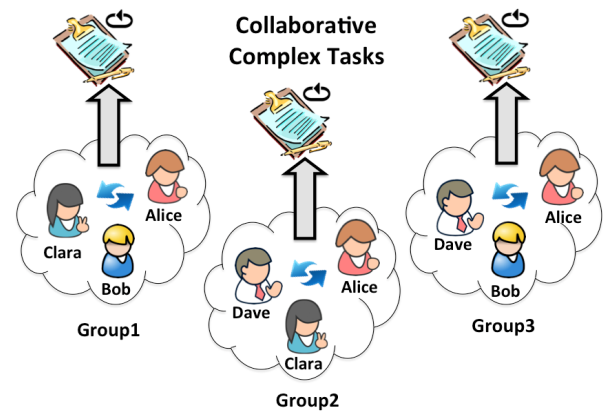


Figure 1: Collaborative document-writing is an instance of complex tasks that require participants to work in groups. They iteratively add, edit and build upon efforts of other, thus making estimation of individual contributions hard. To reward participants individually, there is a need to identify workers from idlers in these groups – the problem we address in this paper.

writing, editing etc. In such scenarios, the requester can evaluate the entire group as a whole (by grading the final article) but cannot gauge individual contributions. If the task-requester is interested in making individual payments based on their respective efforts in the group, she will need a strategy to discriminate between participants. It is a non-trivial task to distinguish workers (who contribute positively) from idlers (who do not contribute to group task) among the participants using only group's performance. Due to lack of concrete evaluation mechanisms, participants may also tend to idle through tasks (Kim and Walker 1984). Prior work also suggests that groups with more than required participants will have idlers (Kenna and Berche 2012).

The task requester is faced with the problem of rewarding individuals based on their group's performance. In this work, we study the problem of distinguishing workers from idlers, without assuming any prior knowledge of individual skills and using only the group output. In this work, a *group* is considered as the smallest observable unit for evaluation and not individual participants. We describe methods to form groups for different collaborative tasks and estimate

individual qualities based on group performances. Further, we give bounds over minimum number of groups required to identify quality of subsets of individuals with high confidence.

We draw upon literature from group testing which proposes strategies for forming groups and mechanisms to decode individual qualities from group results (Atia and Saligrama 2012). These group testing inspired techniques will be used to form groups for collaborative tasks and determine their individual participant qualities based on entire group's performance. These methods, which have traditionally been used for signal processing, automated inspection etc., usually require large number of group tests. Direct application of these algorithms to crowdsourcing would incur huge costs to the task requester. We reduce the total number of group-tasks required for crowdsourcing applications by identifying subsets of workers and idlers.

Our contributions in this paper are:

- We introduce ideas from group testing in crowdsourcing and use them to distinguish workers from idlers, using the performance of their teams.
- Propose efficient extensions, backed with theoretical guarantees, to group testing algorithms for identifying subsets of workers and idlers while significantly reducing the number of required tasks.
- Experiments to empirically validate the theoretical guarantees and show that the proposed algorithms are effective in identifying subset of workers.

Our work underscores interaction between applied crowdsourcing and theoretical foundations of group testing. To our knowledge, we are the first to address the problem of identifying workers from idlers with groups being the smallest unit of evaluation.

In rest of this paper, we first place this work in the context of current literature and then formalize our setting to establish notations. We then present algorithms to classify *all* participants in ideal and noisy scenario and later prove bounds for identifying subsets of worker and idlers. Finally, we validate our theory with experiments and discuss their results.

Related Work

Collaborative Tasks such as article writing, local reporting, product design etc., have recently gained popularity in realms of crowdsourcing (Rahman et al. 2015b; Agapie, Teevan, and Monroy-Hernández 2015). It is known that such tasks require greater efforts and are better accomplished if done in teams (Kittur et al. 2011). On-demand groups of workers can be hired from companies like TaskRabbit for tasks that require physical presence (Teodoro et al. 2014).

Frameworks have been proposed to break down complex tasks into series of simpler tasks (Kulkarni, Can, and Hartmann 2011; Little et al. 2009; Bernstein et al. 2015) and employ teams of crowd workers in tandem. Generating event reports from groups of local workers (Agapie, Teevan, and Monroy-Hernández 2015), is another instance of positive synergistic effects of collaboration (Hertel and Hertel 2011).

Groupsourcing (Chamberlain 2014) investigates the potential of teams on social-networks for solving tasks. However, it assumes that participants are intrinsically motivated and self-organized, which is much different from our setting. (Rahman et al. 2015a) studies the benefit of collaboration in sentence translation and highlights its necessity. However, it assumes the knowledge of worker's skills.

Quality Estimation in Collaborative Tasks. Most work on collaborative tasks look into quality of the result of the tasks. To model collaboration and analyze individual contribution in a group, (Rahman et al. 2015c) use aggregation sum and max strategies. However, we do not assume any prior information of individual skill levels or task expertise requirement. (Eppstein, Goodrich, and Hirschberg 2013) presents combinatorial pair testing to identify unproductive participants working in teams of size exactly two. Our work does not place any restriction over the size of groups. The *free rider problem* (Kim and Walker 1984) in economics presents game-theoretic solutions to similar concerns where idling members, who contribute little toward their groups, still enjoy benefits as any other working member.

Group Testing is a discipline of probabilistic and combinatorial mathematics with rich literature on problems of locating individuals in a set with certain properties by testing them in groups (Du and Hwang 1999). '*Test*' refers to verifying the existence of any identifiable property, in our case - testing whether an individual is a worker or idler or testing whether a team has atleast one slacker.

The idea of group testing is explained with a toy-example below. Suppose we are given samples of several water-bodies to test for the presence of a particular pollutant. A naive method would be to test each sample individually. Instead we can combine a few samples together and test for presence of the pollutant in this pool. Group testing framework gives strategies to mix subset of samples and methods to identify which samples are polluted. This greatly reduces the total number of tests to be performed, and the gains are especially substantial when number of polluted samples are much lesser compared to non-polluted.

In our case, we want to *test* whether a participant is worker or idler by evaluating performance of teams he/she has been a part of. (Dorfman 1943) proposed group tests for identifying diseased individuals by fusing blood samples and testing them in pools. Group testing has found application in numerous other fields such as data mining (Macula and Popyack 2004), multi-access channels (Wolf 1985), data stream (Cormode and Muthukrishnan 2005). Algorithms in Group testing are broadly categorized as *combinatorial* (Ngo and Du 2000) or *probabilistic* (Cheraghchi et al. 2011) based on their methods of forming groups for tests. Most prior works in group testing have focused on identifying properties of *all* items in their universe (Atia and Saligrama 2012) and very little work has also gone into identifying subsets of items. (Sharma and Murthy 2015) have looked into identification of subset from majority class using fewer tests and bound the number of required groups. Our work looks into selecting subsets from both majority and minority classes.

Problem Formulation

In this section we motivate our problem using the example of ‘collaborative science journalism’. Later we establish notations used in further sections and formalize our problem.

Motivating Problem: Consider the content-manager of a collaborative-science journal who has to write several scientific articles regularly. This complex job involves reading recent scientific papers, understanding their findings, focused web-search, summarizing essential details in comprehensible manner etc. The efforts involved per article are prohibitively high for an individual worker. The task-requester can solicit a large pool of crowd-workers and distribute articles among groups of participants. As the members of groups collectively work to create, edit, and publish the content, their individual contributions are fuzzy and hard to track. Participants are given the liberty to collaborate without any supervision and hence are ‘invisible’ (in terms of individual contributions toward group) to the task requester. Being strategic and aware of this limitation in quality assessment, participants can choose to not contribute towards the team-effort. Based on the quality of final article, the requester can determine whether sufficient efforts have gone into producing the article or not. Even if the requester realizes that a few (one or more) participants might have been idling in their groups, she still cannot pinpoint them exactly. The requester is posed with two problems:

- How to smartly form teams for different articles? And
- How to estimate individual participant’s quality based on team performance?

A natural response to the above problem would be to build tracking mechanisms (like version-control) or structuring the task to elicit worker’s accountability. However, isolating intricate cooperation-based-efforts of participants would still be hard as each participant’s impact on the group’s performance is a function of his individual contribution and synergistic cooperation with other subsets of members.

Notations: The requester has access to a set of people \mathcal{N} , with $|\mathcal{N}| = N$. The participants in \mathcal{N} are either *workers* \mathcal{N}_W or *idlers* \mathcal{N}_I ¹ with the number of idlers restricted to K , i.e., $|\mathcal{N}_I| \leq K$. We also assume that idlers are fewer than workers, $K < (N - K)$. There are M collaborative tasks and each task i needs a group of participants $\mathcal{G}_i \subset \mathcal{N}$ to work together toward its completion. Participation matrix $X \in \mathbb{R}^{M \times N}$ captures the information of members for each group task, with $X_{ij} = 1$ if participant j is involved in solving task i , $X_{ij} = 0$ otherwise.

Evaluation function f , which represents the requester, gauges the performance of group \mathcal{G} for a certain task and grades it either satisfactory or not, $f(\mathcal{G}) \rightarrow \{0, 1\}$. $f(\mathcal{G}) = 1$ implies that team performance is not satisfactory and that it consists of one or more idlers, whereas $f(\mathcal{G}) = 0$ implies that team performance is satisfactory and the team is composed of all workers². $Y \in \mathbb{R}^M$ stores the performance of

¹We acknowledge that skill of a person is more fine grained than binary classification of worker or idler, but this simplification is assumed as a first approach.

²This assumption is relaxed later for more realistic settings.

all groups, such that $Y^{(l)} = f(\mathcal{G}_l)$ for l^{th} task.

Noisy Variants: The above strong assumptions made on the nature of f are relaxed in their noisy variants, which are more suitable for crowdsourcing. Due to inherent difficulty of the task or poor judgment of evaluator f , it is possible that task-requester incorrectly evaluates a group \mathcal{G}_A that consists of all workers (and no idler) and assigns it a score $f(\mathcal{G}_A) = 1$. We account for such *false-positives* using parameter q , which is the probability that a group of all workers is evaluated wrong. Similarly, it is possible that few idlers go unnoticed in a group \mathcal{G}_B of diligent workers who compensate for the deficient efforts and $f(\mathcal{G}_B) = 0$. Such *false-negatives* are accounted using parameter u , which is the probability that an idler goes unnoticed while evaluating group \mathcal{G} .

In this work, we present non-adaptive probabilistic algorithms (Cheraghchi et al. 2011) to model groups among crowd workers. These probabilistic methods imitate the randomness in forming groups with unknown workers³. Non-adaptivity requires us to specify all groups upfront, making tasks independent of each other. The requester will not have to wait for results of previous groups to form further groups. Hence, all tasks can happen parallelly or at worker’s availability/convenience. Our algorithms are agnostic to whether participants are distributed/collocated or tools used by them and will prove effective as long as performance of the entire group is computable.

Quality Estimation of All Participants

In this section, we present algorithms to identify *each* participant as either a worker or an idler i.e., partition \mathcal{N} into \mathcal{N}_I and \mathcal{N}_W with $|\mathcal{N}_I| = K$ and $|\mathcal{N}_W| = N - K$. These algorithm (1) assign participants to groups of different tasks; and (2) estimate individual’s quality using performance of entire team. Our goal is to distinguish workers and idlers as soon as possible using minimum number of group tasks.

Noiseless Group Tests

Noiseless setting assumes that the requester can accurately judge a group’s performance and determine if it has performed sub-optimally i.e, there exists one or more idlers in the group. Formally, for a group \mathcal{G}_A , $f(\mathcal{G}_A) = 1 \iff \exists i \in \mathcal{G}_A$ such that $i \in \mathcal{N}_I$ and $f(\mathcal{G}_A) = 0$ otherwise. Although this assumption is strong for collaborative-journalism, it is more applicable to group-sensing scenarios (Singla and Krause 2013) where crowd-workers carrying hand held sensors help in collecting data. Privacy concerns often limit access to individual sensor feed and provide aggregated statistics over data (Olson, Grudin, and Horvitz 2005). In such cases, the requester can accurately identify discrepancies in aggregated data from a group of sensors but cannot pinpoint malicious/defective sensors.

Coupon Collector (CoCo) algorithm is inspired by classical *coupon collection problem* (Chan et al. 2014; Boneh

³Note that random group formation restricts us from modeling participant’s personal preferences to be grouped with other specific participants.

and Hofri 1997). Given a bag of different coupons, where probability of drawing each coupon is the same, coupon collection problem estimates the distribution over total number of draws that are needed to collect all the coupons. Here, we treat each worker $j \in \mathcal{N}_{\mathcal{W}}$ as a coupon and estimate the number of groups j would have to be present in for being identified.

Steps involved in CoCo algorithm: To construct the participation matrix $X \in \mathbb{R}^{M \times N}$, for each row i in X , uniformly sample index j from $[1, 2, 3, \dots, N]$ for g times with replacement and set $X_{ij} = 1$. $g = (\log(N/N - K))^{-1}$ is a system parameter (refer appendix for exact calculation of g) and M is number of group tasks. Note that a few indices might be selected repeatedly, so each row can have less than g ones. Formally, $\forall i, X_{ij} = 1$ if $j \in \{\text{RandomSample}(1, 2, \dots, N)\}^g$. Using this X matrix we assign participants to each i^{th} task and evaluate its team's performance $Y^{(i)}$, to obtain vector Y .

To identify workers from idlers, we look for those groups \mathcal{G} with evaluation $f(\mathcal{G}) = 0$ and mark all their participants as workers. All the remaining participants are declared as idlers. Formally, $\forall i$, if $Y^{(i)} = 0$ then $\forall j$, such that $X_{ij} = 1$ mark $j \in \mathcal{N}_{\mathcal{W}}$ and $\mathcal{N}_{\mathcal{I}} = \mathcal{N} \setminus \mathcal{N}_{\mathcal{W}}$.

It is important to correctly identify workers and idlers with high probability. The contention lies in having sufficiently high M i.e., enough number of team-tasks, so that each worker appears with high probability in atleast one team which does not have idlers. For $g = 1/\ln(N/N - K)$ and $M \geq \theta(K \log N)$, (Chan et al. 2014) bounds the probability of error to diminish exponentially with increase in M .

Noisy Group Tests

Noisy algorithms cater to most crowdsourcing scenarios where an idler does not get identified amidst several workers and workers wrongly get classified as idlers. Let q be the probability that a team of *workers* is wrongly classified and u the probability that an idler does not get identified in group of workers. The requester would still want to correctly identify workers from idlers with high probability.

Column Based (CoBa) algorithm in group testing attempts to correlate columns of X , $X_{:,j}$ with the evaluation vector Y (Atia and Saligrama 2012). For every participant $j \in \mathcal{N}$, it calculates over the number of teams j has been a part of and their respective evaluations.

Steps involved in CoBa algorithm: Construction of participation matrix $X \in \mathbb{R}^{M \times N}$ is as follows. Each participant j is involved with team i with probability p (where parameter $p = 1/K$, refer Appendix for calculation). Formally, $X_{ij} = \mathfrak{B}(p) \forall i, j$, where $\mathfrak{B}(p) = 1$ with probability p and 0 with probability $1 - p$, is Bernoulli distribution. Hence, choosing a participant for any team is an independent event, which reinforces our assumption that we do not have any prior knowledge of individual participants. Using the above participation matrix X , the requester can assign groups for each task and evaluate their performance to obtain Y .

Define function $t(j)$ for participant j such that $t(j) = X_{(:,j)}^T Y' - \psi X_{(:,j)}^T Y$, where Y' refers to complement of Y and ψ is a constant dependent on u, q, K . We

Algorithm 1 Red-CoCo: Reduced Coupon Collection algorithm for identifying subset of workers

Require: M : number of tasks, N : total number of participants, g : parameter

- 1: INITIALIZEMATRIX $X_{ij} = 0, \forall i, \forall j$
- 2: **for all** $i \in [1, 2, 3, \dots, M]$ **do**
- 3: **for** g times **do**
- 4: $j = \text{RANDOMSAMPLE}([1, 2, 3, \dots, N])$
- 5: $X_{ij} = 1$
- 6: **end for**
- 7: **end for**
- 8: $Y = \text{EVALUATEGROUPTASKS}$
- 9: $\mathcal{S}_{\mathcal{W}} = \text{NULL}$
- 10: **for all** $i \in [1, 2, 3, \dots, M]$ **do**
- 11: **if** $Y^{(i)} == 0$ **then**
- 12: $\mathcal{S}_{\mathcal{W}} = \mathcal{S}_{\mathcal{W}} \cup j$, all j such that $X_{ij} = 1$
- 13: **end if**
- 14: **end for**
- 15: **return** $\mathcal{S}_{\mathcal{W}}$

identify a participant j as a worker if

$$t(j) \geq Kc_0/N$$

where c_0 is a constant for fixed u, q, K such that $c_0 = \max_{a,b}[t(a) - t(b)]$. Otherwise, j is identified as an idler.

Intuitively, $X_{(:,j)}^T Y'$ refers to the number of successful groups j has been a part of and $\psi X_{(:,j)}^T Y$ accounts for noisy evaluations. CoBa tends to classify those j 's as workers which are associated with most number of successful teams after accounting for noise. The trade-off here is again between the number of tasks it would take to identify workers correctly with high probability. (Atia and Saligrama 2012) bounds minimum number of tasks M to be $O\left(\frac{K \log N}{(1-u)^2(1-q)}\right)$.

Quality Estimation of Sub-Sets of Participants

In this section, we present algorithms to identify subsets of workers $\mathcal{S}_{\mathcal{W}} \subset \mathcal{N}_{\mathcal{W}}$ and subsets of idlers $\mathcal{S}_{\mathcal{I}} \subset \mathcal{N}_{\mathcal{I}}$ with reduced number of group tasks. It is one of our main contributions to extend the algorithms from previous section and present their analysis for choosing subsets $\mathcal{S}_{\mathcal{I}}, \mathcal{S}_{\mathcal{W}}$.

To motivate the subset selection problem from empirical perspective, consider a case where the collaborative-journal editor hires around 250 participants, out of which he expects approximately 8 – 10% to be idlers. Using CoBa with noisy tests, he will require results for about 500 articles before he can classify all workers and idlers. It is not practical to wait for completion of 500 articles before identifying idlers. The requester can instead choose to identify *a few* workers, who he can pay incentives, and identify *a few* idlers, who he can drop from further tasks. The requirement of classifying all participants is now reduced to classifying a subset of participants.

The crux of group testing algorithms is to have each participant take part in sufficiently many group tasks, so that we can assert his worker/idler status with high certainty.

To classify *all* participants as either worker or idler it takes $M = O(K \log N)$ for noisy and noiseless scenarios (Chan et al. 2014), which we improve next.

Subsets in Noiseless Group Tests

In coupon collection problem it takes lesser number of draws to collect initial coupons while it takes long time to collect last few coupons in coupon-collection problem. For instance, first coupon is always collected in the very first draw whereas it takes N number of draws (in expectation) to collect the N^{th} coupon, given all other $(N - 1)$ coupons are collected. We know from previous sections that CoCo identifies all $(N - K)$ workers when each is grouped in atleast one team having no idlers and it takes $M_{\text{CoCo}} = O(K \log N)$ group tasks.

Reduced Coupon Collector (Red-CoCo) algorithm reduces M by exploiting the above property of coupon collection problem while selecting subset of workers $S_{\mathcal{W}} \subset \mathcal{N}_{\mathcal{W}}$ with $|S_{\mathcal{W}}| = L$. We show that Red-CoCo takes $O(K \log(N - L))$ fewer tasks to accomplish this requirement. The reduction comes from the fact that now only L workers need to be grouped in team with no other idler.

Steps involved in Red-CoCo algorithm: Algorithm 1 describes the construction of participation matrix for Reduced-CoCo and method for distinguishing individual workers. This method is same as CoCo and only varies in the input parameter M . Red-CoCo considers only those tests for which $Y^{(i)} = 0$, hence it is limited to identifying only subset of workers and fails to identify idlers from remaining $N - L$ participants. As the task requester would want to identify workers as soon as possible, the theorem below formalizes the reduction in number of group tasks to identify $S_{\mathcal{W}}$ with $L < (N - K)$.

Theorem 1. *For fixed N and K , Red-CoCo requires atleast M tasks, with*

$$M \geq K \log \left(\frac{N}{N-L} \right)$$

for identifying subset of L workers $S_{\mathcal{W}} \subset \mathcal{N}_{\mathcal{W}}$ with probability of error decaying exponentially with M .

Note that the bound on number of tasks M is $O(K \log(N - L))$ fewer as compared to CoCo. The proof follows from calculating the number of draws required to collect first L coupon (workers) from total $N - K$ coupon (workers). We upper bound the probability of error by estimating tail-probabilities for not drawing atleast L coupons in $N \log N / (N - L)$ draws. Refer Appendix for detailed proof and parameter assignments.

Subsets in Noisy Group Tests

Similar to noiseless scenario, identifying all workers and idlers in noisy setting takes considerably more tasks than identifying subsets of participants. Note, we assume that workers in \mathcal{N} are in majority i.e., number of workers exceed the number of idlers $N - K > K$. In group testing, subset identification in noisier settings has been previously explored in (Sharma and Murthy 2015), where subset is restricted to the majority class (workers in our case). Analysis in (Sharma 2014) shows reduction in the number of group

Algorithm 2 Red-CoBa: Reduced Column based algorithm for identifying subset of workers and subset of idlers

Require: M : number of tasks, N : total number of participants, p : parameter, L : worker subset size, D : idler subset size, ψ : parameter

- 1: **for all** $i \in [1, 2, 3, \dots, M]$ **do**
- 2: **for all** $j \in [1, 2, 3, \dots, N]$ **do**
- 3: $X_{ij} = \mathfrak{B}(p)$
- 4: **end for**
- 5: **end for**
- 6: $Y = \text{EVALUATEGROUPTASKS}$
- 7: **for all** $j \in [1, 2, 3, \dots, N]$ **do**
- 8: $\mathcal{T}(j) = X_{:,j}^T Y' - \psi(X_{:,j}^T Y)$
- 9: **end for**
- 10: $S = \text{SORT}(\mathcal{T})$
- 11: $S_{\mathcal{W}} = S[N - L + 1 : N]$
- 12: $S_{\mathcal{I}} = S[1 : D]$
- 13: **return** $S_{\mathcal{W}}$ and $S_{\mathcal{I}}$

tests from $\theta(K \log N)$ to $O(KL/N)$ for identifying a subset of size L from majority class.

Reduced Column Based (Red-CoBa) algorithm extend this analysis to identify subsets in minority class (idlers in our case) along with identifying subset of workers simultaneously. Note that finding $S_{\mathcal{W}}$ and $S_{\mathcal{I}}$ together makes up for the deficiency of Red-CoCo which can only identify $S_{\mathcal{W}}$.

Steps involved in Red-CoBa algorithm: Participation Matrix X is constructed in a similar way as for CoBa. For identifying L workers, we sort each participant j on basis of $\mathcal{T}(j) = X_{:,j}^T Y' - \psi(X_{:,j}^T Y)$, where Y' denotes complement of Y and ψ is constant dependent on u, q, K . Top L high scored participants are declared as workers and bottom D are declared idlers. (Sharma and Murthy 2015) show that for

$$M_{\text{CoBa}} \geq \frac{K}{\lambda'} \left(\frac{\log \left[K \binom{N-K}{L-1} \right]}{N - K - (L - 1)} \right) \quad (1)$$

where $\lambda' = c_3(1 - \gamma)^2(1 + \psi)(1 - q)/(1 - u)$ the probability of error in selecting L workers decays exponentially in M (refer appendix for parameter details). Algorithm 2 demonstrates step-by-step method for Red-CoBa.

We use M from Equation (1) above to find workers subset $|S_{\mathcal{W}}| = L$ and also additionally identify subset of idlers $|S_{\mathcal{I}}| = D$. We give upper bounds on maximum number of idlers D that can be identified with high probability, using M tests. Note, no additional tests are performed in Red-CoBa to find idlers separately. To identify most number of idlers for given M group tasks, the following theorem formalizes the upper-bound on D .

Theorem 2. *For M satisfying Equation (1) and fixed N and K , Red-CoBa (Algorithm 2) can identify at most D idlers with arbitrarily high probability $1 - \exp(-Mc_1)$, for some constant $c_1 > 0$, where*

$$D \leq (K + 1) - \frac{\log(N - K - L)}{\log \left(K \binom{N-K}{L-1} \right)} (N - K - (L - 1))$$

Proof follows from upper bounding the probability of mis-identifying a worker as an idler. We calculate probability that

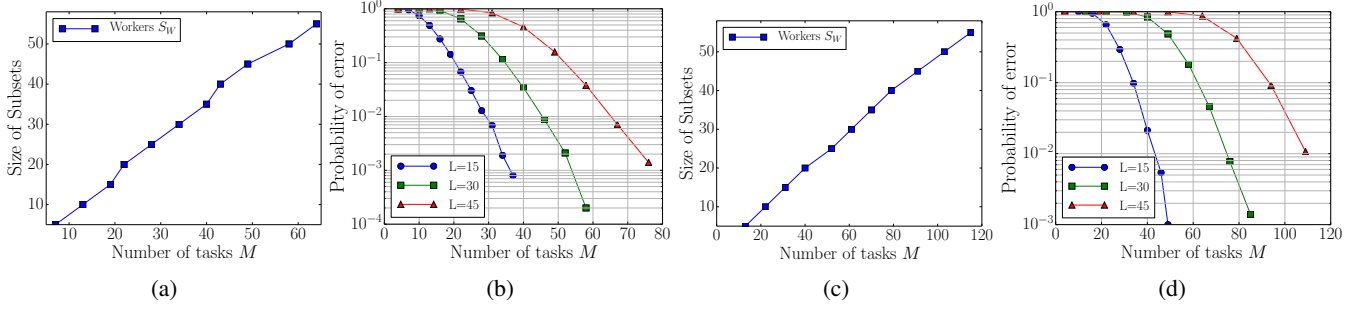


Figure 2: Performance of Red-CoCo in identifying subsets of workers \mathcal{S}_W (note that idlers are not identified here) and empirical validation of Theorem 1. (a) For $N = 125$, $K = 25$, size of subsets of workers L against number of group tasks M needed. (b) For $N = 125$, $K = 25$, probability of error diminishes exponentially with increasing M . (c) For $N = 250$, $K = 100$, L against number of group tasks needed M for extracting $|\mathcal{N}_W| = L$. (d) For $N = 250$, $K = 100$, exponential decay of probability of error.

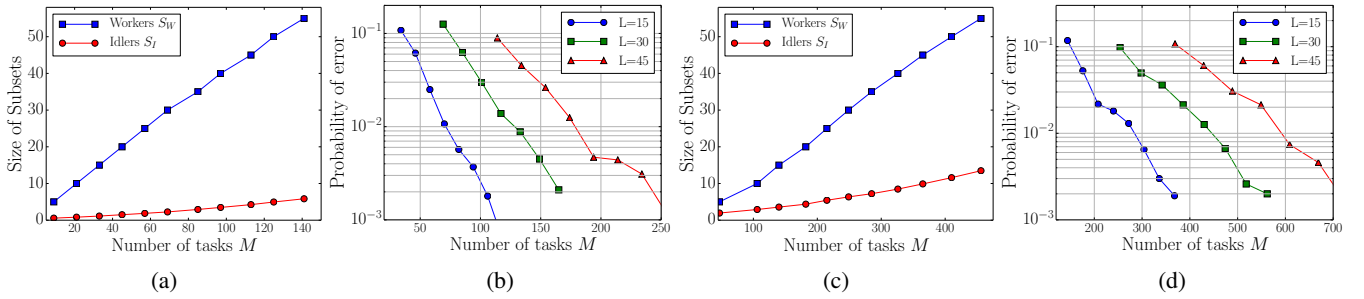


Figure 3: Performance of Red-CoBa in identifying subsets of workers \mathcal{S}_W and subset of idlers \mathcal{S}_I and empirical validation of Theorem 2. (a), (b), (c) and (d) are respective results of Red-CoBa over identical scenarios from Figure 2.

$\mathcal{T}(j) \leq \mathcal{T}(i)$ for worker j and idler i in sorted vector \mathcal{T} . Taking union over all such events $\forall i, \forall j$ and calculate for D such that error decays exponentially. Refer Appendix for detailed proof and parameter calculations.

Experiments

We conduct two sets of experiments on simulated data in this section. The first set aims to give empirical support to the claims made in our theorems with varying size of workers \mathcal{S}_W and idlers \mathcal{S}_I . The second set of experiments try to learn the impact of crowd properties like proportion of idlers vs. workers in \mathcal{N} , behavior of noise in team evaluation etc.

We ask the following questions in this sections:

- *Q1*: How many workers and idlers can be identified with significant probability for a given number of group-tasks?
- *Q2*: For fixed size of workers/idlers, does the probability of error in mis-identification decay exponentially with increase in number of group-tasks?
- *Q3*: How does participant demographics and noise impact the size of subsets of workers/idlers?

Note, for verifying performance of Red-CoCo and Red-CoBa over real crowd workers, their true worker/idler status is needed, which is hard to obtain. We simulate datasets

that closely imitate crowd scenarios to test the merit of our proposed algorithms.

Setup: We experiment with two different participant populations. \mathcal{N}_1 consists of total $N = 125$ participants with $K = 25$ of them as idlers. It has the generic distribution of 80%:20% for workers vs. idlers. \mathcal{N}_2 with total participants $N = 250$ consisting of $K = 100$ idlers. This is another prevalent crowd scenario where the requester hires many workers (potentially consisting of many volunteers) and several of them slack in group-efforts. We set noise parameters as $u = 0.05$ and $q = 0.1$. The results are averaged over 10000 repetitions and we consider events with probability greater than 0.9 to have significant chances of occurrence.

Method: Participation matrices are constructed as described in Algorithm 1 and Algorithm 2. K random participants from total N population are assigned *idler* identity and used to calculate group performance Y . For noiseless setting, the evaluation vector is

$$Y = \bigvee X_{:,j} \quad \forall j \in \mathcal{N}_I$$

with logical-OR of columns of all idlers. In noisy setting, where an idler can go unnoticed with probability u and team of workers can get mis-evaluated with probability q , the group performance vector is calculated by

$$Y = \left(\bigvee \mathbb{U}_j X_{:,j} \right) \bigvee Q \quad \forall j \in \mathcal{N}_I$$

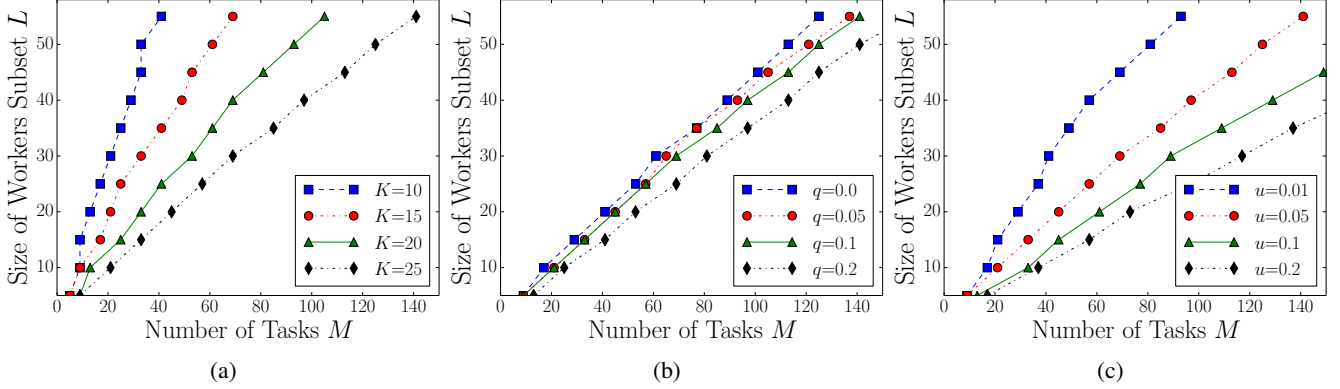


Figure 4: Effect of crowd demographics and noise parameters on size of subset of workers identified against number of group tasks. (a) For $N = 125$, varying the number of idlers K in total population. It requires least number of tasks to identify $\mathcal{S}_{\mathcal{W}}$ when number of idlers are less. (b) For $N = 125K = 25$, varying probability q of mis-identifying workers. Observe that it is less harmful in terms to number of tasks M to mis-classify workers. (c) For $N = 125K = 25$, varying probability u of not identifying an idler. Observe that it takes many more tasks to uncover idlers hidden in groups of workers.

where matrix \mathbb{U}_j hides idlers j in task i with probability u and vector Q mis-evaluates each group with probability q . Precisely, $\mathbb{U}_j \in \mathbb{R}^{M \times M}$ is diagonalized matrix of vector $U_j \in \mathbb{R}^M, U_j^{(i)} = \mathfrak{B}(1 - u)$ and $Q \in \mathbb{R}^M, Q^{(i)} = \mathfrak{B}(q)$. From this point, the true identity of participants is hidden from the algorithms and we would like them to be able to predict these true identities.

Results: We are not aware of other crowd evaluation techniques which consider *groups of participants* as their smallest observable unit. We present the outcome of our algorithms and answer the above three questions.

Q1 Size of subsets identified: Figure 2 and 3 show the performance of Red-CoCo and Red-CoBa respectively, over \mathcal{N}_1 and \mathcal{N}_2 . Figure 2a, 2c and Figure 3a, 3c capture the dependence between number of workers/idlers that can be identified with varying number of group tasks. Although there are fewer workers in $|\mathcal{N}_{\mathcal{W}1}| = 100$ in comparison to $|\mathcal{N}_{\mathcal{W}2}| = 150$, it takes many more group tasks for \mathcal{N}_2 than \mathcal{N}_1 to arrive at same size of subsets of workers. This is because groups \mathcal{G} in \mathcal{N}_2 have higher probability of having idlers and hence lower probability of $f(\mathcal{G}) = 0$. It highlights the idea that gains in group tests are much higher when $|\mathcal{N}_{\mathcal{W}}| \gg |\mathcal{N}_{\mathcal{I}}|$. It is also evident from Figure 3a and 3c that number of idlers identified is much lesser than number of workers $|\mathcal{S}_{\mathcal{W}}| > |\mathcal{S}_{\mathcal{I}}|$, which is consistent with the our theory.

Q2 Probability of error: For different fixed values of L , Figure 2b, 2d and Figure 3b, 3d show that probability of error i.e., mis-identifying worker as idler and vice versa, decreases exponentially to linear increase in number of group tasks. This indicates that the requester can become much more confident about his judgment of identifying workers from idlers by conducting a few additional group tasks.

Q3 Impact of parameters: With other parameters fixed, Figure 4a shows that it takes more number of group tasks to identify workers with increase in number of idlers. It is be-

cause the probability of $f(\mathcal{G}) = 0$ drops with increasing K for random groups \mathcal{G} . Similarly, Figure 4b and 4c show that increase in noise levels of the system leads to more number of group tasks for identifying $\mathcal{S}_{\mathcal{W}}$. Comparing results across Figure 4a, 4b and 4c shows that idlers getting away unidentified in pool of workers is a more serious problem than mis-evaluating workers. It take more efforts to pinpoint an idler who piggybacks over efforts of workers in his group. It also shows that the best way of reducing group tasks to identify subsets, is to start with a population of lesser K .

Discussion on Relevance and Limitations

Wider Relevance of Group Tests in Crowdsourcing: Although we motivated this work from the perspective of ‘collaborative-science-journal’, it could also applicable to other complex tasks like book translations, local media reporting, group-sensing, group pedagogy etc. Crowdsourcing applications which satisfy the following two prime requisites can benefit from this framework. First, the task should involve *diffused* efforts from groups of people, i.e it should not be easy to identify individual’s contributions. Second, there should be a provision to evaluate entire group as a whole.

Group testing in general can have much wider applicability in crowdsourcing. Algorithms can be utilized to solve mainstream crowdsourcing micro-tasks. A task requester can pool a few micro-tasks together and ask the crowd for its judgment over entire pool. Based on the results of several such pools, the requester can deduce information about individual micro-tasks. This framework guides on how to group individual entities of interest and further deduce information about them. By means of this work, we hope to highlight that Crowdsourcing applications can benefit from Group Testing and Compressed-Sensing algorithms at large.

Simplicity is a strength of both Red-CoCo and Red-CoBa algorithms and they are computationally very inexpensive.

Creating participation matrix requires access to a random number generator and the requester can manually work out participant qualities with minimal efforts.

Limitations of this model: As a first approach to apply group testing in crowdsourcing, there are a few shortcomings in the current model and they require non-trivial extensions. We presently classify participants as either workers or idlers, whereas in reality the skills of a person are more nuanced and should be measured on a richer scale. Second, our underlying assumption that a worker remains to be a worker and idler continues as idler across group tasks, is hard to satisfy in crowdsourcing. Crowd workers are generally dynamic and get motivated/demotivated by incentives. Accounting for their dynamic identities might add to the randomness of our system. And lastly, our random allocation of groups, although unbiased, ignores the complementary effects of collaboration. A participants productivity might vary with different groups (maybe collaborate better with friends than strangers). Our model currently fails to identify such inter-participant dependencies and would require extending it to adaptive group test strategies.

Conclusion

Collaborative crowdsourcing is an emerging field where group of participants coherently work to accomplish complex tasks. With machine-intelligence continuously gaining competence to solve simpler tasks, crowd-intelligence is going to be increasingly used for complex tasks which will be better accomplished by groups rather than individuals.

We addressed the problem of distinguishing workers from idlers in scenarios where requester has no knowledge of individual contributions but only entire group's performance. We proposed efficient extensions to group testing based algorithms to decide on how to form teams and identify subsets of individuals with reduced number of tasks. We gave theoretical guarantee over the performance of algorithms and conducted simulation based experiments to validate them empirically. As part of future work, we want to account for skills of people and incorporate parameters for dynamic participants who shift from being workers to idlers and vice versa.

Acknowledgments

We thank the anonymous reviewers for their constructive comments and Chandramouli Kamanchi for useful discussions. We also thank Uday Saini, Danish and Rohit Vaish for carefully reading drafts of this paper.

References

Agapie, E.; Teevan, J.; and Monroy-Hernández, A. 2015. Crowdsourcing in the field: A case study using local crowds for event reporting. In *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2015, November 8-11, 2015, San Diego, California*, 2–11.

Atia, G. K., and Saligrama, V. 2012. Boolean compressed sensing and noisy group testing. *Information Theory, IEEE Transactions on* 58(3):1880–1901.

Bernstein, M. S.; Little, G.; Miller, R. C.; Hartmann, B.; Ackerman, M. S.; Karger, D. R.; Crowell, D.; and Panovich, K. 2015. Soylent: a word processor with a crowd inside. *Communications of the ACM* 58(8):85–94.

Boneh, A., and Hofri, M. 1997. The coupon-collector problem revisited: a survey of engineering problems and computational methods. *Stochastic Models* 13(1):39–66.

Chamberlain, J. 2014. Groupsourcing: Distributed problem solving using social networks. In *Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014, November 2-4, 2014, Pittsburgh, Pennsylvania, USA*.

Chan, C. L.; Jaggi, S.; Saligrama, V.; and Agnihotri, S. 2014. Non-adaptive group testing: Explicit bounds and novel algorithms. *IEEE Trans. Information Theory* 60(5):3019–3035.

Cheraghchi, M.; Hormati, A.; Karbasi, A.; and Vetterli, M. 2011. Group testing with probabilistic tests: Theory, design and application. *Information Theory, IEEE Transactions on* 57(10):7057–7067.

Cormode, G., and Muthukrishnan, S. 2005. What's hot and what's not: tracking most frequent items dynamically. *ACM Transactions on Database Systems (TODS)* 30(1):249–278.

Dorfman, R. 1943. The detection of defective members of large populations. *The Annals of Mathematical Statistics* 14(4):436–440.

Du, D.-Z., and Hwang, F. K. 1999. *Combinatorial group testing and its applications*, volume 12. World Scientific.

Eppstein, D.; Goodrich, M. T.; and Hirschberg, D. S. 2013. Combinatorial pair testing: distinguishing workers from slackers. In *Algorithms and Data Structures*. Springer. 316–327.

Hertel, G., and Hertel, G. 2011. Synergetic effects in working teams. *Journal of Managerial Psychology* 26(3):176–184.

Kenna, R., and Berche, B. 2012. Managing research quality: critical mass and optimal academic research group size. *IMA Journal of Management Mathematics* 23(2):195–207.

Kim, O., and Walker, M. 1984. The free rider problem: Experimental evidence. *Public Choice* 43(1):3–24.

Kittur, A.; Smus, B.; Khamkar, S.; and Kraut, R. E. 2011. Crowdforge: crowdsourcing complex work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, October 16-19, 2011*, 43–52.

Kulkarni, A. P.; Can, M.; and Hartmann, B. 2011. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, 2053–2058. ACM.

Little, G.; Chilton, L. B.; Goldman, M.; and Miller, R. C. 2009. TurkIt: tools for iterative tasks on mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, 29–30. ACM.

Macula, A. J., and Popyack, L. J. 2004. A group testing method for finding patterns in data. *Discrete applied mathematics* 144(1):149–157.

Ngo, H. Q., and Du, D.-Z. 2000. A survey on combinatorial group testing algorithms with applications to dna library screening. *Discrete mathematical problems with medical applications* 55:171–182.

Olson, J. S.; Grudin, J.; and Horvitz, E. 2005. A study of preferences for sharing and privacy. In *CHI'05 extended abstracts on Human factors in computing systems*, 1985–1988. ACM.

Rahman, H.; Roy, S. B.; Thirumuruganathan, S.; Amer-Yahia, S.; and Das, G. 2015a. Task assignment optimization in collaborative crowdsourcing. In *Data Mining (ICDM), 2015 IEEE International Conference on*, 949–954. IEEE.

Rahman, H.; Roy, S. B.; Thirumuruganathan, S.; Amer-Yahia, S.; and Das, G. 2015b. "the whole is greater than the sum of its parts": Optimization in collaborative crowdsourcing. *CoRR* abs/1502.05106.

Rahman, H.; Thirumuruganathan, S.; Roy, S. B.; Amer-Yahia, S.; and Das, G. 2015c. Worker skill estimation in team-based tasks. *PVLDB* 8(11):1142–1153.

Sharma, A., and Murthy, C. R. 2015. On finding a subset of non-defective items from a large population using group tests: Recovery algorithms and bounds. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 4150–4154.

Sharma, A. 2014. *Finding a Subset of Non-defective Items from a Large Population: Fundamental Limits and Efficient Algorithms*. Ph.D. Dissertation, Indian Institute of Science, Bangalore.

Singla, A., and Krause, A. 2013. Incentives for privacy tradeoff in community sensing. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Teodoro, R.; Ozturk, P.; Naaman, M.; Mason, W.; and Lindqvist, J. 2014. The motivations and experiences of the on-demand mobile workforce. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, 236–247. ACM.

Wolf, J. K. 1985. Born again group testing: Multiaccess communications. *Information Theory, IEEE Transactions on* 31(2):185–191.

Appendix

Red-CoCo treats the problem of selecting $S_{\mathcal{W}}$ as collecting L coupons from total $N - K$ coupons. Before we proceed to proving Theorem 1 we establish a few essential properties of coupon collection problem. For total n coupons, the probability of drawing a new coupon, having collected i coupons already, is $(n - i + 1)/n$. The expected time drawing i^{th} coupon is $t_i = n/(n - i + 1)$. Expected time for collecting l coupons out of total n coupons

$$\begin{aligned} \mathbb{E}(T_{n-l}) &= \mathbb{E}(t_1) + \mathbb{E}(t_2) \dots + \mathbb{E}(t_l) \\ &= \frac{n}{n} + \frac{n}{n-1} + \dots + \frac{n}{n-l+1} \\ &= n(\log n - \log(n-l)) \end{aligned}$$

Probability that i^{th} coupon is not collected in r draws is $(1 - 1/n)^r \leq e^{-r/n}$. Substituting $r = \beta n \log(n/n - l)$, $e^{-r/n} = (n/n - l)^{-\beta}$. By union bounds, probability that it takes more than $n \log(n/n - l)$ draws to pick l coupons is $n(n/n - l)^{-\beta}$.

Proof of Theorem 1. Consider the process of drawing Mg random participants to construct participation matrix X , as coupon collection problem where probability of drawing each participant from $X_{i,:}$ is uniform. For some group \mathcal{G} in Red-CoCo, the probability that $f(\mathcal{G}) = 0$ is $((N - K)/K)^g$, i.e all g participants are workers. As Red-CoCo considers only those groups for which $Y^{(l)} = 0$, the expected number of participants in all such rows is $Mg((N - K)/N)^g$. To find values of M for which at least L workers are present participants of $X_{i,:}$ is

$$(1 - \epsilon)Mg \left(\frac{N - K}{N} \right)^g \geq \beta(N - K) \log \left(\frac{N - K}{N - K - L} \right) \quad (2)$$

Left side of Equation (2) represents expected number of workers to be found in row $X_{i,:}$ with $Y^{(l)} = 0$ and right side denotes the event of exceeding the expected number of draws to obtain L workers. Inequality in Equation (2) hold with probability atleast

$$1 - \exp \left(-\epsilon^2 M \left(\frac{N - K}{N} \right)^g \right) - (N - K) \left(\frac{N - K}{N - K - L} \right)^{-\beta} \quad (3)$$

which is obtained using Chernoff bound on left side of Equation (2) and tail bound of coupon collection problem on its right side, which is proved above (substitute $n = (N - K)$). Differentiating Equation (2) with respect to g and by substituting back optimal $g = 1/\log(N/N - K)$, we get

$$M \geq \frac{e\beta}{(1 - \epsilon)} (N - K) \log \left(\frac{N - K}{N - K - L} \right) \log \left(\frac{N}{N - K} \right)$$

Simplifying $\log(N/(N - K))$ using inequality $\log(1 + x) \geq x - x^2/2$ for $x = (K/(N - K))$, we get

$$\begin{aligned} &\frac{e\beta}{(1 - \epsilon)} (N - K) \log \left(\frac{N - K}{N - K - L} \right) \left(\frac{K}{N - K} - \frac{K^2}{(N - K)^2} \right) \\ &\geq \frac{e\beta}{(1 - \epsilon)} \log \left(\frac{N - K}{N - K - L} \right) K \\ &\geq \eta K \log(N - K) - K \log(N - K - L) \text{ for appropriate } \eta \\ &\geq \eta K \log(N) - K \log(N - L) \end{aligned}$$

for sufficiently large population N and relatively less idlers $K = o(N)$. Hence, $M_{RedCoCo} = M_{CoCo} - K \log(N - L)$.

For fixed parameters N, K, L the probability of error by substituting optimal $g = 1/\log(N/N - K)$ in complement of Equation (3) is

$$\mathbb{P}(\mathcal{E}_{err}) = \exp \left(-\frac{\epsilon^2 M}{e} \right) + (N - K) \left(\frac{N - K}{N - K - L} \right)^{-\beta}$$

□

In **Red-CoBa**, the probability of participant j , not being selected in group i is $up + (1 - p)$. Probability that $f(\mathcal{G}_i) = 0$ for any group i^{th} is when none of the K idlers participate in it and workers do not get misclassified. Say, $\tau = \mathbb{P}(Y^{(l)} = 0) = (1 - q)(1 - (1 - u)p)^K$.

Given that an idler is present in a group \mathcal{G} , then $f(\mathcal{G}) = 0$ when the idler gets lost among other workers, no other idler participates

in $f(\mathcal{G})$ and workers do not get mis-identified.

$$\begin{aligned}
\mathbb{P}(Y^{(l)} = 0 | X_{li} = 1) &= \gamma\tau, \quad i \in \mathcal{N}_{\mathcal{I}} \\
\gamma &= \frac{u}{1 - (1-u)p} \\
\mathbb{P}(Y^{(l)} = 0 | X_{li} = 0) &= (1 - (1-u)p)^{K-1} (1-q) \\
\mathbb{P}(X_{li} = 0 | Y^{(l)} = 0) &= \gamma p \\
\psi &= \frac{\tau(1+\gamma)}{(1-p)} \\
p &= 1/K
\end{aligned} \tag{4}$$

Proof of Theorem 2. In Red-CoBa (Algorithm 2) the first D participants from sorted \mathcal{T} are declared as idlers. Error occurs when a worker is part of $S_{\mathcal{I}}$, i.e. $\mathcal{T}(j)$ for some worker j is lesser than atleast $K - (D - 1)$ idlers. Let S_Z denote the set of workers remaining after choosing $S_{\mathcal{W}}$, with $|S_Z| = N - K - L$. Let S_D denote subset of $K - (D - 1)$ idlers and $S_{\mathcal{D}}$ denote set of all valid S_D , so $|S_D| = K - D + 1$ and $|S_{\mathcal{D}}| = \binom{K}{D-1}$. More formally, error occurs when $\mathcal{T}(j), j \in S_Z$ is less than all $\mathcal{T}(i), i \in S_D$. Let \mathcal{E}_d denote the event of error while selecting D idlers, i.e. Red-CoBa returns $S_{\mathcal{I}}$ with one or more workers.

$$\begin{aligned}
\mathcal{E}_D &\subset \bigcup_{j \in S_Z} \{j \in S_{\mathcal{I}}\} \\
&\subset \bigcup_{j \in S_Z} \bigcup_{S_D \in \mathcal{S}_{\mathcal{D}}} \{\mathcal{T}(j) \leq \mathcal{T}(i), \forall i \in S_D\}
\end{aligned}$$

Define \mathbb{P}_d as the probability of event $(\mathcal{T}(j) - \mathcal{T}(i) \leq 0; i \in S_D, j \in S_Z)$ for given idler and worker pair. By union bounds

$$\mathbb{P}(\mathcal{E}_D) \leq (N - K - L) \binom{K}{D-1} (\mathbb{P}_d)^{K-(D-1)}$$

Rearrange $\mathcal{T}(j) - \mathcal{T}(i)$ as $(X_{:,j}^T Y' - X_{:,i}^T Y') + \psi(X_{:,i}^T Y - X_{:,j}^T Y)$, and rewriting

$$\begin{aligned}
&= \underbrace{\left(\sum_{l=1}^M X_{:,j}^{(l)} - X_{:,i}^{(l)} \right) \mathbb{I}_{Y^{(l)}=0}}_{Z_0} + \psi \underbrace{\left(\sum_{l=1}^M X_{:,i}^{(l)} - X_{:,j}^{(l)} \right) \mathbb{I}_{Y^{(l)}=1}}_{Z_1} \\
&= Z_0 + \psi Z_1 = Z
\end{aligned}$$

where $X_{:,j}^{(l)}$ denotes l^{th} component of $X_{:,j}$ and \mathbb{I} is indicator function. The idea is to compute probability of event $(Z < 0)$ by finding mean μ_Z and variance σ_Z^2 of Z . By using Equation (4) and Bayes rule, we get

$$\begin{aligned}
\mathbb{E}(Z_1) &= \mathbb{E} \left[\left(\sum_{l=1}^M X_{:,i}^{(l)} - X_{:,j}^{(l)} \right) \mathbb{I}_{Y^{(l)}=1} \right] \\
&= M(1 - \tau) \left(\mathbb{P}(X_{li} = 1 | Y^{(l)} = 1) \right. \\
&\quad \left. - \mathbb{P}(X_{lj} = 1 | Y^{(l)} = 1) \right) \\
&= M(1 - \tau) \left(\frac{p(1 - \gamma\tau)}{1 - \tau} - p \right) \\
&= M\tau(p - \gamma p)
\end{aligned}$$

Similarly, computing expectation and variance of other random variables

$$\begin{aligned}
\mathbb{E}(Z_0) &= M\tau(p - \gamma p) \\
\mathbb{E}(Z_0^2) &= M\tau p(1 - 2\gamma p + \gamma) \\
\mathbb{E}(Z_1^2) &= Mp[(1-p)(2 - \gamma\tau) + \tau(1 - \gamma p)] \\
\mu_Z &= \mathbb{E}(Z_0) + \psi \mathbb{E}(Z_1) = M\tau p(1 + \psi)(1 - \gamma) \\
\sigma_Z^2 &= \sigma_{Z_0}^2 + \psi^2 \sigma_{Z_1}^2 \leq Mp[\tau(1 + \gamma) + 2\psi^2(1 - p)]
\end{aligned}$$

For arbitrarily small ϵ , calculate probability $\mathbb{P}_d = \mathbb{P}(Z < 0) \leq \mathbb{P}(Z < \epsilon\mu_Z)$ using Bernstein inequality and take union bound to get simplified $\mathbb{P}_d \leq \exp(-M\lambda)$ where $\lambda = p\tau^3(1 - \gamma^2)(1 + \gamma)(1 - \epsilon)^2$.

Substituting upper bounds for $\psi = \frac{\tau(1+\gamma)}{(1-p)}, p = 1/K, \lambda' = K\lambda$,

$$\begin{aligned}
\mathbb{P}_d &\leq \exp\left(-\frac{M}{K}\lambda'\right) \\
\mathbb{P}(\mathcal{E}_d) &\leq \exp\left(-\frac{M\lambda'}{K/(K-D+1)}\right. \\
&\quad \left. + \ln\left[(N-K-L)\binom{K}{D-1}\right]\right) \\
&= \exp\left(-M\left(\frac{\lambda'}{K/(K-D+1)}\right.\right. \\
&\quad \left.\left. - \frac{\ln\left[(N-K-L)\binom{K}{D-1}\right]}{M}\right)\right) \\
&= \exp(-Mc_1)
\end{aligned}$$

where $c_1 \geq 0$ for exponential decay of probability of error. Substituting M from (Sharma and Murthy 2015)

$$M \geq \frac{K}{\lambda'} \left(\frac{\ln \left[K \binom{N-K}{L-1} \right]}{(N-K)-(L-1)} \right)$$

for $c_1 > 0$, we obtain

$$\begin{aligned}
\underbrace{\frac{\ln \left[K \binom{N-K}{L-1} \right]}{(N-K)-(L-1)}}_{\Delta} &\geq \frac{\ln \left[(N-K-L) \binom{K}{D-1} \right]}{K-(D-1)} \\
e^{\Delta(K-D+1)} &\geq (N-K-L) \binom{K}{D-1} \\
\frac{e^{\Delta K}}{N-K-L} &\geq \binom{K}{D-1} e^{\Delta(D-1)} \tag{5}
\end{aligned}$$

$$\begin{aligned}
\binom{K}{D} &= \frac{K \cdot (K-1) \dots (D+2) \cdot (D+1)}{(K-D) \cdot (K-D-1) \dots 2 \cdot 1} \\
&= \frac{1}{\left(1 - \frac{D}{K}\right) \left(1 - \frac{D}{K-1}\right) \dots \left(1 - \frac{D}{(D+1)}\right)} \\
&\geq \left(1 - \frac{D}{K}\right)^{-(K-D)}
\end{aligned}$$

in limits, for K and $D = o(K)$, above inequality approximate to e^D by substituting $x = -D/K$ in $(1 + \frac{x}{n})^n$ converging to e^x . Substituting in Equation (5) $e^{\Delta K} \geq (N-K-L)e^{\Delta(D-1)+D}$

$$\begin{aligned}
\frac{K\Delta - \log(N-K-L) + \Delta}{\Delta} &\geq D \\
(K+1) - \frac{\log(N-K-L)}{\log\left(K \binom{N-K}{L-1}\right)} (N-K-(L-1)) &\geq D
\end{aligned}$$

□