An Incentive Compatible Multi-Armed-Bandit Crowdsourcing Mechanism with Quality Assurance

Shweta Jain, Sujit Gujar, Satyanath Bhat, Onno Zoeter, Y. Narahari

June 30, 2014

Abstract

Consider a requester who wishes to crowdsource a series of identical binary labeling tasks from a pool of workers so as to achieve an assured accuracy for each task, in a cost optimal way. The workers are heterogeneous with unknown but fixed qualities and moreover their costs are private. The problem is to select an optimal subset of the workers to work on each task so that the outcome obtained from aggregating labels from them guarantees a target accuracy. This problem is challenging because the requester not only has to learn the qualities of the workers but also elicit their true costs. We develop a novel multi-armed bandit (MAB) mechanism for solving this problem. We propose a framework, Assured Accuracy Bandit (AAB), which leads to an adaptive, exploration separated MAB algorithm, Strategic Constrained Confidence Bound (CCB-S). We derive an upper bound on the number of exploration steps which depends on the target accuracy and true qualities. We show that our CCB-S algorithm produces an ex-post monotone allocation rule which can be transformed into an ex-post incentive compatible and ex-post individually rational mechanism that learns qualities of the workers and guarantees the target accuracy in a cost optimal way.

1 Introduction

Consider a company that provides financial advice to a series of clients on whether to invest in a particular security or not. In order to provide such advice to each client, the company has a pool of financial consultants. Gathering the opinion of as many consultants as possible and aggregating their opinions (for example, using majority voting) increases the probability of providing a high accuracy advice, however, it also entails increased costs. The company has two conflicting business requirements, firstly to keep the costs low, and secondly, to provide an advice that meets a minimum threshold accuracy. The individual financial consultants have heterogeneous but unknown skill sets (qualities) and their costs are typically private information. Since the consultants are strategic, they might report higher costs for their services. On the other hand, to meet an accuracy threshold, the company needs to learn the qualities of the consultants. The problem addressed in this paper is motivated by such real world problems that are quite common in the settings of crowdsourcing and expert-sourcing.

As an abstraction to such problems, we consider a series of binary labeling tasks. Consider a setting where the requester has a pool of workers with different qualities which are unknown. In addition, the costs of the workers are their private information and thus can be strategically misreported. As a design objective, the requester, given the noisy labels from the set of workers, requires that the final outcome obtained by aggregating the labels from the selected workers achieves a certain target accuracy. The target accuracy parameter provides a handle on the tradeoff between cost and accuracy. A high value of target accuracy increases the probability of getting the right answer but at the same time it may call for a larger number of workers to be commissioned which can raise the costs. Therefore, the requester can choose a suitable target accuracy as per the task sensitivity at hand. In a nutshell, the goal of the requester is to select a subset of workers to achieve a desired accuracy level of the aggregated label for each task with
minimum cost, at the same time giving the right incentives to the workers so that they report their costs truthfully.

In the non-strategic machine learning version of the problem, the costs are known or equivalently, the workers report costs truthfully. The requester has to select an efficient set of workers so as to minimize the costs while learning the qualities. Though the requester can learn the qualities of the workers over a period by observing their performance on similar tasks, overusage of the workers may incur significant costs. Thus, the requester faces a dilemma of exploration (where he has to learn the qualities of the workers) versus exploitation (where he has to choose the workers optimally based on learnt qualities). A natural solution to this problem can be designed using techniques developed for the multi-armed bandit (MAB) problem [3]. However, an important challenge in our settings is, we need to ensure the accuracy constraint which in turn depends on unknown qualities. Thus, there is a need to develop a new framework to address the problem.

An interesting challenge arises when the costs of the workers are private and the strategic workers try to manipulate the learning algorithm by mis-reporting their costs so as to benefit themselves. We refer to this as strategic version of the above problem, where we have the additional task of eliciting the true costs using a suitable mechanism. If qualities were known, a natural way to ensure truthfulness is to use the benchmark VCG (Vickery-Clarke-Groves) mechanism. Since qualities are not known and need to be learnt, a VCG mechanism cannot be applied directly [3]. Thus, we need to solve two problems simultaneously. In short, we need to marry techniques from machine learning with game theory that would ensure truthful behaviour of the workers while learning the qualities. Often such mechanisms are referred as Multi-Armed Bandit Mechanisms or simply MAB mechanisms [6, 13, 25].

Learning qualities in non-strategic settings as above has been proposed in [28] using MAB techniques, [19] proposed a primal dual technique whereas [24] proposed an expectation maximization (EM) based algorithm for the same. However, none of these works addresses the game theoretic setting as in our paper.

1.1 Contributions

The above discussion clearly highlights the need for designing a new framework for the problem of selecting cost optimal, strategic workers so as to achieve a target accuracy. Hence there is a need for a quality assuring mechanism that would select a cost-optimal subset of workers, whose qualities are unknown and costs are private information. This paper fills this important research gap for binary labeling tasks by modeling this problem in the multi-armed bandit framework. We propose a new framework for such problems, which we call Assured Accuracy Bandit (AAB). We consider two versions of this problem: (1) non-strategic version where the costs are known and the qualities of the workers have to be learnt and (2) strategic version where the costs are to be truthfully elicited as well. In particular, the following are our contributions in this paper.

- We propose a novel framework, Assured Accuracy Bandit (AAB), in which we formulate an optimization problem, where the goal is to minimize the cost subject to the constraint that error probability is below a certain threshold level with respect to true qualities for a binary labeling task. As pointed out above, we consider two versions of AAB.

Non-Strategic Version

- In this setting, we design an adaptive exploration separated algorithm, which we call, Non-Strategic Constrained Confidence Bound (CCB-NS).

- Though the true qualities are not known, our algorithm makes sure that the constraint is satisfied with high probability (Theorem 4.1).
We provide an upper bound on the number of exploration steps for a given problem that depends on the target accuracy and true qualities (Theorem 4.2).

**Strategic Version**

- In the strategic version of this problem where workers may not report their costs truthfully, we present an elegant modification of the algorithm CCB-NS which we call strategic constrained confidence bound (CCB-S) and prove that the allocation rule provided by CCB-S is ex-post monotone (Theorem 5.2) in terms of cost.

- Given this ex-post monotone allocation rule, we adapt the techniques from [5] to design an ex-post truthful and ex-post individually rational mechanism (Corollary 5.3).

We extend the theory of MAB mechanisms in a novel way to design an ex-post incentive compatible and ex-post individually rational mechanism for this setting. MAB mechanisms are popular in sponsored search auctions which are forward auctions. We extend the work to crowdsourcing in particular to our model which belongs to a reverse (or procurement) auction setting. To the best of our knowledge, this is the first mechanism that learns qualities of strategic workers who have costs as private information.

### 1.1.1 Organization

The paper is organized as follows. We present a summary of relevant work in Section 2. In Section 3, we provide a general formulation of the problem. Next, we present our model in two different stages. First, we discuss the non-strategic model in Section 4 and next in Section 5, we discuss the strategic version using mechanism design. Future work and conclusions are provided in Section 7.

### 2 Related Work

First, we describe the state of the art pertaining to improving accuracy of predicted answer by learning qualities of workers. We will then look into the mechanism design literature in crowdsourcing. Our setting is more general as it involves both learning and mechanism design. MAB mechanisms provide a natural solution in such setting. We then review some of the MAB problems and MAB mechanism design literature.

#### Learning in Crowdsourcing

Ho et. al. [19] considered a very similar setting as of ours where an assured quality needs to be satisfied for each task. However, uniform and known cost of the workers are assumed and a specific error probability function is taken into consideration. We address heterogeneous setting with costs being privately held by strategic workers and we work with general error probability function rather than a specific error probability function with weighted majority as in theirs. Abraham et. al. [1] consider a setting where a certain accuracy is required to be met for a given micro-task. The authors considered the problem of aggregating answers in a sequential way until a certain accuracy is achieved. Homogeneous workers are assumed in a cluster and thus the goal is to select a single optimal crowd for a single task. In a general setting, their assumption of a crowd having sufficient number of homogeneous quality workers may not hold. Our setting is more general where a subset of optimal workers (arms) needs to be selected with heterogeneous qualities at one go for a given micro-task and thus the problem maps to multiple pull multi-armed bandit problem. A similar approach of improving quality of answers while minimizing the cost is considered by Karger et al. [21] where the final answer is predicted using a low rank approximation method. Work by [24] considers learning a classifier while learning
the qualities of the workers using EM [12] algorithm. In a similar line of work, Viappiani et al. [29] consider a Bayesian approach to learn the class label by taking noisy observations from experts. However, the models proposed in [29, 12, 24] work well experimentally, there are no analytical guarantees on the predicted outcome. Tran-Thanh et al. [28] present an algorithm for efficient selection of workers based on MAB algorithms by formulating a knapsack problem. For each task, a single non-strategic worker is selected as opposed to our work where subset selection of strategic workers is considered. However, to enhance the accuracy of binary labeling tasks, multiple noisy answers are needed from the workers.

Though the literature addresses how to learn the quality of workers, none of the above paper addresses the challenge in meeting the target accuracy on each task and does not consider strategic version where costs can be misreported by the workers.

**Mechanism Design in Crowdsourcing**

Various works on mechanism design in crowdsourcing involve designing pricing strategies with online workers. Babaioff et al. [4] use an MAB mechanism to determine an optimal pricing mechanism for a crowdsourcing problem having homogeneous qualities within a specified budget (known as bandits with knapsack). Work by Singla and Krause [27] assumes costs to be private information and proposes a posted price mechanism to elicit true costs from the users using MAB mechanisms while maintaining a budget constraint. Other works involving mechanism design in online procurement auctions are considered in [7, 26]. The work in [4, 27, 7, 26] considers homogeneous quality workers as opposed to heterogeneous quality of ours. Our setting is more general where an auction mechanism is considered to elicit heterogeneous costs from the workers.

Work by Garg et al. [16] and Bhat et al. [8] considers cost of the workers to be public and qualities to be private strategic quantity of the workers. Another line of work involves incentivizing people to work with their true qualities, when qualities are privately held by the workers [30] in peer prediction markets whereas [10] analyze crowdsourcing tasks as winner take it all auctions in game theoretic settings. They assume that only one worker gets paid and do not try to learn the qualities over period.

Current literature either proposes a posted price mechanism or addresses the problem of eliciting qualities. Our work is more general where we learn the qualities over a period of time, at the same time eliciting true costs.

**MAB Algorithms**

A rich body of literature is available on the MAB problem. We are concerned with the stochastic MAB setting, where rewards of each arm are fixed but unknown. A recent survey by Bubeck and Cesa-Bianchi [9] compiles various variations on stochastic and non-stochastic MAB problem. The closest setting as ours is considered by Shipra Agrawal and Nikhil Devanur [2] where a general bandit problem with concave rewards and convex constraints is solved. Our problem is more general, as the constraints need not be convex. Moreover, the constraint is satisfied on an average in [2] as opposed to our work, where the constraint needs to be satisfied at each round. The probably Approximately Correct (PAC) learning framework for single pull and multiple pull MAB is considered by Even Dar et al. [15] and by Kalyanakrishnan et al. [20] respectively. Our learning algorithm may appear closely related to the PAC learning setting. However, the solution obtained from PAC algorithms is approximately correct with high probability after arms are pulled for a certain number of rounds, which depends on the provided approximation factor and confidence. In our setting, the goal is to select an optimal set with high probability since a constraint needs to be satisfied with respect to stochastic qualities. The combinatorial MAB problem introduced by Chen, Wang, and Yuan [11] is relevant to our work, however, they deal with an unconstrained setting. A constrained MAB problem for single pull is discussed by Ding et. al. [14], where each arm is associated with random rewards and the goal of the
algorithm is to maximize the reward such that total cost which is also stochastic in all rounds does not exceed budget but the constraint is on overall rounds instead of each round.

MAB Mechanisms

Multi-armed Bandit mechanisms in the forward setting, in particular, in sponsored search auction are recent advancements that combine the area of MAB problems and mechanism design. The important result provided by [6, 13] says that any deterministically truthful MAB mechanism must be exploration separated i.e. allocation in the learning phase should not depend on the bids and thus the regret of any such algorithm is at least $O(T^{2/3})$ where $T$ is the total number of rounds. Work in [17, 25] extends the result to multiple pull multiarmed bandits i.e. to the case of multiple slot sponsored search auction. The techniques developed in these papers cannot be adopted to our setting because 1) workers need to be paid in spite of their failure as opposed to the setting where payment is made only if there is a success 2) ours is the constrained multi-armed bandit setting as opposed to the setting of traditional multi-armed setting where a subset of best arms is selected without constraints. Also note that, ours is a reverse auction setting as opposed to the forward auction in the existing literature on MAB mechanisms. Babaioff et al. [5] design a general procedure which takes any monotone allocation rule as input and converts it into a randomized truthful mechanism which implements the input allocation rule with high probability and requires evaluation of the input allocation rule exactly once. As an application of this transformation, an MAB mechanism that is ex-post incentive compatible and ex-post individual rational with regret of $O(T^{1/2})$ is proposed. In our current work, we use this transformation and propose an ex-post monotone allocation rule in the case of a reverse auction in a constrained multi-armed bandit setting.

Our very preliminary results appeared as [18] as an extended abstract. However, this paper is significantly more general. Only a particular error probability function was considered in [18] to ensure target accuracy. In this paper, we consider a class of error probability functions satisfying monotonicity and bounded smoothness properties which we define later.

3 The Model

Let $N$ be the set of $n$ crowdsourcing workers, who are available for completing $T$ similar crowdsourcing tasks. Each agent or worker $i$ has an associated quality $q_i \in [0,1]$, which represents the probability of giving the correct answer by him. By similar tasks, we mean that each worker’s quality is same for all the tasks. We assume that the workers are not spammers and their quality of service is at least 0.5. The quality of any worker $i$ is assumed to be independent of the qualities of the other workers. There is a cost $c_i \in [0,1]$ associated with each worker that can be reported strategically by the workers. Without loss of generality, we assume that costs of the workers lie between 0 and 1. Let $1 - \alpha$ be the target accuracy ($\alpha$ is the threshold level) provided by the requester that determines the tradeoff between cost and accuracy to be achieved for a particular task. We consider binary classification tasks where labels are either zero or one. Our model is summarized in Figure 1.

Notations are summarized in Table 1. The error on a task with inputs from workers depends on the qualities of the workers and the rule to aggregate these answers. We abstract this as error probability function which we describe in the following subsection.

3.1 Error Probability Function

Let $f_S(q)$ be any function that represents the measure of error probability ($1 - f_S(q)$ represents the accuracy) when set $S$ is selected with quality profile $q = (q_1, q_2, \ldots, q_n)$. The problem we seek to solve in this paper involves minimizing the cost at the same time satisfying the constraint that $f_S(q) < \alpha$ where $(1 - \alpha)$ is the target accuracy. Depending on aggregation rules
and the requester requirements, different error probability function can exists. Our framework and solution approach is general which works with any error probability function that satisfies following properties:

3.1.1 Properties of an error probability function

- **Monotonicity**: $f_S(q)$ is said to be monotone if for all quality profiles $q$ and $q'$ such that

  $\forall i \in \mathcal{N} \; q'_i \leq q_i$ we have, $f_S(q') < \alpha \iff f_S(q) < \alpha \forall S \subseteq \mathcal{N}, \forall \alpha \in [0, 1]$ 

  that is increase in quality of each worker can only increase the accuracy or decrease the error probability.

- **Bounded smoothness**: $f_S(q)$ satisfies bounded smoothness property if there exists a monotone continuous function $h$ such that if

  $\max_i |q_i - q'_i| \leq \delta \implies |f_S(q) - f_S(q')| \leq h(\delta) \forall S \subseteq \mathcal{N}, \forall q, q' \in [0.5, 1]$ 

  This property ensures that effect in accuracy with slight perturbation in quality is bounded by a continuous function $h$.

These properties are similar to the properties satisfied by the reward function in [11] and are satisfied by various error probability functions. Next, we give certain examples of error probability functions that satisfy the properties of monotonicity and bounded smoothness when majority voting is used as an aggregation rule. Note that, the algorithm is general enough to incorporate any aggregation rule and any error probability function if monotonicity and bounded smoothness properties are satisfied.

3.1.2 Examples of error probability functions

Let $S$ be the selected set with players $\{1, 2, \ldots, s\}$ with quality profile $q_i$ such that $q_1 \leq q_2 \leq \ldots \leq q_s$ to whom we assign the task $t$. For notational convenience, we drop $t$ and let $\tilde{y}_i$ be
Table 1: Notation Table

| Notation | Description |
|----------|-------------|
| \( N \)  | Set of workers available |
| \( n \)  | Number of workers available |
| \( S \)  | Set of workers selected |
| \( T \)  | Number of tasks |
| \( \alpha \) | \((1 - \alpha)\) is the target accuracy required |
| \( \mu \) | Confidence level required to satisfy constraint |
| \( y_t \) | True label for task \( t \) |
| \( \tilde{y}_i^t(S) \) | Vector of noisy labels of set \( S \) |
| \( \hat{y}_t \) | Predicted label for task \( t \) |
| \( n_{i,t} \) | Number of times worker \( i \) is selected till \( t \) number of tasks |
| \( q_i \) | True quality of worker \( i \) |
| \( \hat{q}_i(t) \) | Estimated quality of worker \( i \) till \( t \) number of tasks |
| \( \hat{q}_i^+(t) \) | Upper confidence bound (UCB) on \( q_i \) till \( t \) number of tasks |
| \( \hat{q}_i^-(t) \) | Lower confidence bound (LCB) on \( q_i \) till \( t \) number of tasks |
| \( c_i \) | True cost of worker \( i \) |
| \( c_i \) | True cost of worker \( i \) |
| \( c_{-i} \) | Cost vector of all the workers except \( i \) \((c_1, c_2, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n)\) |
| \( c \) | Cost vector \((c_1, c_2, \ldots, c_n)\) |
| \( C(S) \) | \( \sum_{i \in S} c_i \) |
| \( \hat{c}_i \) | Reported cost of worker \( i \) |
| \( S^* \) | Optimal set with respect to true qualities |
| \( S^t \) | Selected set for task \( t \) |
| \( f_S(q) \) | Error probability function of \( S \) with \( q \) |
| \( \Delta \) | Separation of \( f_S(q) \) from \( \alpha \) for all set i.e. \(|\alpha - f_S(q)| > \Delta \ \forall S\) |
| \( \hat{C} \) | Cost incurred when constraint is not satisfied |
| \( R \) | Reward given when constraint is satisfied |

the noisy label that we get from the player \( i \in \{1, 2, \ldots, s\} \) and \( \tilde{y}(S) = \{\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_s\} \) be the vector of noisy labels from worker’s set \( S \). Then, the predicted label \( \hat{y} \) when majority voting rule is used as an aggregation rule is given by:

\[
\hat{y} = \begin{cases} 
1, & \text{if } \sum_{i \in S} \tilde{y}_i > s/2 \\
0, & \text{otherwise.}
\end{cases}
\]

1) Probability of most likely outcome that leads to an error is given by [18]:

\[
P(E_{S(q)}) = \max_{\hat{y}(S) \in \{0,1\}^S} P(\hat{y}(S), \hat{y} \neq y | y) \\
= \max_{\hat{y}(S) \in \{0,1\}^S} \left( P(\hat{y} \neq y | y, \hat{y}(S))P(\hat{y}(S) | y) \right) \\
= (1 - q_1)(1 - q_2) \ldots (1 - q_{s'})q_{s'+1} \ldots q_s, \\
\text{where } s' = \left\lfloor \left( (s + 1)/2 \right) \right\rfloor
\]

where, the last equality comes from the fact that \( P(\hat{y} \neq y | y, \hat{y}(S)) \) can be either zero of one as \( \hat{y}(S) \) is fixed and \( P(\hat{y}(S) | y) \) is maximum when lower half quality workers make mistake in which case predicted label \( \hat{y} \neq y \). Instead of satisfying the constraints with respect to \( P(E_{S(q)}) \),
one can satisfy the constraint with respect to the quantity $\hat{P}(ES(q))$ which is given as:

$$f_S(q) = \hat{P}(ES(q)) = (1 - q_1)(1 - q_2) \cdots (1 - q_s)$$  \hspace{1cm} (1)

Note that $f_S(q)$ satisfies monotonicity as well as bounded smoothness property with $h(\delta) = \frac{n}{2}\delta$ if $0 \leq q_i - \delta \leq 1$ and $0 \leq q_i + \delta \leq 1 \forall i \in N$.

2) Average probability of error is given by [22]:

$$P(ES(q)) = P(y = 1)P(\hat{y} = -1|y = 1) + P(y = -1)P(\hat{y} = 1|y = -1)$$

$$= \pi P\left(\sum_{i=1}^{s} \tilde{y}_i < 0|y = 1\right) + (1 - \pi)P\left(\sum_{i=1}^{s} \tilde{y}_i > 0|y = -1\right)$$

Let us now focus on $P\left(\sum_{i=1}^{s} \tilde{y}_i < 0|y = 1\right)$.

$$E[\tilde{y}_i|y = 1] = -P(\hat{y}_i = -1|y = 1) + P(\hat{y}_i = 1|y = 1) = (2q_i - 1)$$

Now,

$$P\left(\sum_{i=1}^{s} \tilde{y}_i < 0|y = 1\right) = P\left(\sum_{i=1}^{s} \tilde{y}_i - E[\sum_{i=1}^{s} \tilde{y}_i] < -E[\sum_{i=1}^{s} \tilde{y}_i]|y = 1\right)$$

$$= P\left(\sum_{i=1}^{s} \tilde{y}_i - E[\sum_{i=1}^{s} \tilde{y}_i] < -\sum_{i=1}^{s} (2q_i - 1)\right)$$

$$\leq \exp\left(\frac{-\left(\sum_{i=1}^{s} (2q_i - 1)\right)^2}{2\sum_{i=1}^{s} 1}\right) \text{ (By Hoeffding’s inequality)}$$

Similarly, it can be shown that:

$$P\left(\sum_{i=1}^{s} \tilde{y}_i > 0|y = -1\right) \leq \exp\left(\frac{-\left(\sum_{i=1}^{s} (2q_i - 1)\right)^2}{2\sum_{i=1}^{s} 1}\right)$$

Assuming $\pi = 1 - \pi = 0.5$, we get,

$$P(ES(q)) \leq f_S(q) = \exp\left(\frac{-\left(\sum_{i=1}^{s} (2q_i - 1)\right)^2}{2\sum_{i=1}^{s} 1}\right)$$

Again, one can verify that function $f_S(q)$ is monotone and satisfies bounded smoothness property.

From the above two examples we see that, $f_S(q)$ can be made to satisfy both the assumption of monotonicity and bounded smoothness.

Now, We describe our framework in which the optimization problem is posed.
3.2 Assured Accuracy Bandit (AAB) Framework

Recall that, a task \( t \in \{1, \ldots, T\} \) needs to be completed with assured accuracy provided by the requester with optimal cost in a sequential fashion. Thus, for each task \( t \), the goal of the requester is to select a set of workers \( S^t \), such that the error probability function is less than the threshold level. At the same time, the requester has to make sure that the tasks are completed or accomplished optimally in terms of costs. Hence for each round \( t \), we need to solve the following optimization problem.

\[
\min_{X^t_i \in \{0, 1\}} \sum_i c_i X^t_i \\
\text{s.t.} \\
f_{\{i, X^t_i = 1\}}(q) < \alpha
\]

(2)

where the qualities of the workers are not known a priori and hence need to be learned by giving tasks repeatedly to the workers. Also, solving the optimization problem, the requester has to make sure that the constraint in (2) is satisfied with respect to the true qualities with high confidence. We initially select large number of workers so as to meet the accuracy with respect to the true qualities and then move towards selecting the optimal set of workers once enough confidence is gained. We refer this novel framework as Assured Accuracy Bandits.

3.2.1 Regret in AAB Framework

Performance of any MAB algorithm is typically measured by the regret it achieves. Regret in an MAB framework is defined to be the reward difference between the learning algorithm and the optimal algorithm. We will see later that our algorithm is designed in such a way that for each task \( t \), constraint given by (2) is satisfied with probability \((1 - \mu)\), where \( \mu \) is the confidence parameter that can be set by the requester. Thus we can define regret of an algorithm \( A \) if the constraint is satisfied as:

\[
R(A) = \sum_{t=1}^{T} \sum_{i \in S^t} c_i - T \sum_{i \in S^*} c_i
\]

(3)

\( S^t \) is the set selected by the algorithm \( A \) for task \( t \)

\( S^* \) is the optimal set with known qualities

Since the constraint for each task is satisfied with probability \((1 - \mu)\), we can also bound the total expected regret by:

\[
E[R(A)] = (1 - \mu) \left( \sum_{t=1}^{T} \sum_{i \in S^t} c_i - T \sum_{i \in S^*} c_i \right) + \mu \tilde{C} T
\]

where \( \tilde{C} \) is the cost that is incurred by the requester if constraint fails to satisfy.

3.3 Assumptions

Here, we summarize the assumptions we make in our model.

- We consider series of binary classification tasks as an abstraction to our problem. However, the results can be easily generalized.

- We assume that the error probability function satisfies the assumptions of monotonicity and bounded smoothness. These assumptions are very natural and are satisfied by most of the error probability functions.
We assume that the true label is observed once the task is completed. To motivate this assumption, we recall the trading example given in the introduction section where the company and the clients can realize the true label by the end of the day, for example, in intraday trading.

We assume that if all workers are selected, then the constraint is always satisfied with respect to true qualities. Thus, if qualities are only partially learnt, then the algorithm can select the complete set and satisfy the constraint. This is equivalent to saying that there are enough number of workers.

The reward $R$ which the requester gets for satisfying the constraint and the cost $\tilde{C}$ he incurs for not satisfying the constraint are such that it is always beneficial for the requester to satisfy the constraint. This is true in a natural way in typical applications where a heavy penalty may be incurred for providing a wrong answer.

### 3.4 Related MAB Algorithms

There are various MAB algorithms for different related settings. Here we list the two general and most relevant algorithms for our setting.

#### 3.4.1 The UCB Algorithm

The UCB algorithm proposed by Auer et al.\[3\] works for classical MAB problem where only one arm needs to be pulled at every time. This algorithm maintains an upper confidence bound (hence the name UCB) on each arm which depends on its empirical reward as well as on the exploration factor to give the arm enough number of pulls. The algorithm achieves the regret of $O(\ln T)$ and it is the best possible regret that can be achieved. One extension to the multiple pull setting is to pull the arms in the increasing order of their upper confidence bound.

#### 3.4.2 The CUCB Algorithm

CUCB algorithm\[11\] works for more general framework where a set of arms are selected with combinatorial rewards. At each time, a subset of arms are pulled and reward of all the selected arms are revealed. The algorithm works for general non linear reward functions as long as monotonicity and bounded smoothness properties are satisfied by the reward functions. The general idea of the algorithm is to select the subset such that the reward is maximized with respect to the upper confidence bounds of all the arms similar to UCB algorithm where a single arm was selected with highest upper confidence bound. The CUCB algorithm achieves a regret of $O(\ln T)$.

The next section describes the non strategic version of the problem where we assume that costs are public knowledge. Note that, we can bound the overall regret by bounding the number of rounds in which the algorithm selects a sub-optimal set $S'$ i.e. $C(S') > C(S^*)$. We call such rounds *non-optimal rounds*. Note that, if $C(S') = C(S^*)$, then we get zero regret for those rounds with probability $(1 - \mu)$. We then propose an algorithm called Non-Strategic Constrained Confidence Bound (CCB-NS) which selects a constant number of such bad rounds with high confidence. We then provide a modified algorithm Strategic Constrained confidence bound (CCB-S) that for the strategic case.

### 4 Non-Strategic Version

In this setting, our goal is to learn the qualities of the workers while assuming costs to be publicly known. We solve a general optimization problem given by Equation (2) with unknown qualities which can only be learnt over a period of time. Since workers obtain their label according to
the true qualities (though unknown to them), the constraint should be with respect to true qualities. Since these qualities are unknown to the requester, he has to make sure that the constraint is satisfied with high probability.

In our approach, we maintain an upper confidence bound on qualities similar to UCB algorithm [3]. Adapting techniques from [11] [2], the optimization problem is solved using upper confidence bound. This step ensures that the workers not far from optimal set are selected by the algorithm. However, the selected set though satisfying the constraint with respect to upper confidence bound, may fail to satisfy the constraint with respect to true qualities. Thus, we cannot directly return the same solution. The main idea of our algorithm is to maintain the lower confidence bounds also which provide lower bounds on true qualities with high confidence. And the selected set is checked for the constraint with respect to lower confidence bound. If constraint is not satisfied with respect to lower confidence bound, workers are added to the selected set to satisfy the constraint with respect to lower confidence bound. We now present the non strategic version of constrained confidence bound algorithm (CCB-NS) that satisfies the constraint at each round with high probability.

4.1 CCB-NS Algorithm

The CCB-NS algorithm (presented in Algorithm 1) works on the principle of the UCB algorithm [3] and ensures that the constraint in (2) is satisfied with high confidence \( \mu \). Input to the algorithm is parameter \( \alpha \), the target accuracy (which is assumed to be same for all tasks), number of tasks \( T \), number of workers \( n \), and confidence level \( \mu \) with which constraint in (2) is satisfied. Output of the algorithm will be the subset \( S^t \) and predicted label \( \hat{y}_t \) for each task \( t \). The predicted label \( \hat{y}_t \) is decided based on an aggregation function (AGGREGATE) such as majority voting or weighted majority etc. of noisy labels collected from the set \( S^t \). Initially all the workers are selected to have some estimate about the qualities (Step 1). Their noisy labels are aggregated and a label is predicted. Next, the algorithm observes the true label and updates the workers are selected to have some estimate about the qualities (Step 1). Their noisy labels are aggregated and a label is predicted. Next, the algorithm observes the true label and updates the mean quality estimates, upper confidence and lower confidence bounds. Let \( n_{i,t} \) denotes the number of times \( i^{th} \) worker is assigned the task and \( c_{i,t} \) denotes number of times he has provided the correct label till \( t \) number of tasks. Similar to the UCB algorithm [3], algorithm maintains upper confidence and lower confidence bounds on qualities. These bounds are given as follows:

\[
\hat{q}_{i}^+(t) = \hat{q}_i(t) + \sqrt{\frac{1}{2n_{i,t}} \ln \left( \frac{2n}{\mu} \right)}, \quad \hat{q}_{i}^-(t) = \hat{q}_i(t) - \sqrt{\frac{1}{2n_{i,t}} \ln \left( \frac{2n}{\mu} \right)} \text{ where } \hat{q}_i(t) = c_{i,t}/n_{i,t}
\]

By Hoeffding’s inequality one can prove that true quality \( q_i \) lies between \( \hat{q}_{i}^-(t) \) and \( \hat{q}_{i}^+(t) \) with probability \( 1 - \frac{\mu}{n} \) for any task \( t \) and for any worker \( i \). Since true qualities of the workers lie between \([0.5, 1]\), we initialize \( \hat{q}_{i}^+(t) \) and \( \hat{q}_{i}^-(t) \) by 1 and 0 respectively. In the algorithm, we represent \( \hat{q}_i \), \( \hat{q}_{i}^+ \) and \( \hat{q}_{i}^- \) to be the estimates till \( t \) number of tasks for convenience. For the remaining tasks, the optimization problem is first solved with respect to the upper confidence bound. The optimal worker set for task \( t \) is denoted by \( S^t \). Since the constraint might not be satisfied with respect to true qualities, algorithm then checks if selected set also satisfies constraint with respect to lower confidence bound (Step 7), if it satisfies the algorithm simply returns the set \( S^t \) for all the future rounds and this is in fact the optimal set \( S^* \) with probability at least \( 1 - \mu \) (Lemma 4.1). Thus for the remaining rounds, the algorithm simply returns the same set (Step 16) and label \( \hat{y}_t \) accordingly. If the constraint is not satisfied then a minimal set of workers is added to the selected set so as to satisfy the constraint with lower confidence bound (Step 8). If no set satisfies the constraint with lower confidence bound, the algorithm simply returns the complete set \( \mathcal{N} \). By our assumption that complete set \( \mathcal{N} \) satisfies the constraint with respect to true qualities, we can assume that the constraint is being satisfied with respect
to true qualities even though it is violated with respect to the lower confidence bound. Finally, the vectors \( \hat{q}, \hat{q}^+, \hat{q}^- \) are updated (Steps 10).

We first see that the algorithm CCB-NS satisfies the constraint at each round with high probability.

Note that by Hoeffding’s inequality, for each \( i \), \( \hat{q}_i^- \leq q_i \leq \hat{q}_i^+ \) with probability \( 1 - \mu_n \). Thus, by the union bound we have, \( \forall i \in N, \hat{q}_i^- \leq q_i \leq \hat{q}_i^+ \) with probability greater than \( 1 - \mu \). For brevity of notations, in the rest of the paper we will use \( \hat{q}^- \leq q \leq \hat{q}^+ \) to represent \( \forall i \in N, \hat{q}_i^- \leq q_i \leq \hat{q}_i^+ \).

**Theorem 4.1** CCB-NS satisfies the constraint in Equation \( 2 \) with probability at least \( 1 - \mu \) at every round \( t \).

**Proof:**

- By our assumption that if all players are selected, the constraint is always satisfied, thus, in the rounds in which all players are selected, the constraint is satisfied.

- Now, if set \( S^t \) is returned by CCB-NS, then
  \[ f_{S^t}(\hat{q}^-) < \alpha \implies f_{S^t}(q) < \alpha \]  
  with probability \( 1 - \mu \) (By monotonicity. \( \hat{q}^- \leq q \) with probability at least \( 1 - \mu \))

We now show that once the constraint in Step 7 is satisfied then the set \( S_{opt}^t = S^* \) with probability at least \( 1 - \mu \). For simplicity, in the rest of the paper, we assume that there exists a unique optimal set \( S^* \), though the results can be easily generalized to the case where there are multiple optimal sets.

**Lemma 4.1** Set \( S_{opt}^t \) returned by the CCB-NS algorithm is an optimal set with probability (w.p) at least \( 1 - \mu \). That is, \( C(S_{opt}^t) = C(S^*) \) w.p \( 1 - \mu \)

**Proof:** Let \( t^* \) be the round in which CCB stops exploring. At \( t = t^* \),

- \( f_{S^*}(\hat{q}^+) < \alpha \) with probability \( 1 - \mu \) by monotonicity property

- Let \( S_{opt}^t \) be the set of workers selected by CCB-NS. As CCB-NS solves the optimization problem \( 2 \)
  \[ C(S_{opt}^t) \leq C(S^*) \]

At \( t^* \),

\[ f_{S^*}(\hat{q}^-) < \alpha \implies f_{S^*}(q) \leq f_{S^*}(\hat{q}^-) < \alpha \]  
with probability at least \( 1 - \mu \)  
\[ \implies C(S_{opt}^t) = C(S^*) \]  
(By uniqueness assumption)
**Definition 4.1 (Non-optimal Subset)** We say that at round $t$, a set $S^t$ selected by the algorithm is non-optimal subset, if $S^t \neq S^\ast$.

**Definition 4.2 (Non-optimal Round:)** We say a round $t$ is non-optimal round if selected set $S^t$ is not the optimal set $S^\ast$.

**Definition 4.3 ($\Delta$-Separated Property:)** We say that $q$ satisfies $\Delta$-Separated property with respect to threshold $\alpha$ if $\exists \Delta > 0$ such that, $\Delta = \inf_{S \subseteq \mathcal{N}} |f_S(q) - \alpha|$. That is, for no set of workers, $S$, has probability of error $f_S(q) \in [\alpha - \Delta, \alpha + \Delta]$.

Note that the value of $\Delta$ is typically unknown to the requester since qualities are unknown. However, our algorithm does not require the value of $\Delta$ beforehand. We will show that the number of exploration steps depends on the value of $\Delta$, and thus, CCB-NS is adaptive exploration separated. We now bound the number of exploration steps.

**Lemma 4.2** If $\forall i \in \mathcal{N} \ n_{i,t} \geq \frac{2}{(\ln^2 (\Delta / \alpha))} \ln(2n / \mu)$ then for any task $t$

1. for all sets $S \neq S^\ast$, $f_S(q) > \alpha \implies f_S(\hat{q}^+) > \alpha$ with probability $1 - \mu$.
2. \( f_{S^*}(\hat{q}^-) < \alpha \) with probability \( 1 - \mu \)

**Proof:** Let \( l = \frac{2}{(\ln^{-1}(\Delta))^2} \ln\left(\frac{2n}{\mu}\right) \)

- By Hoeffding’s inequality, \( \hat{q}^+_i - q_i \leq 2\sqrt{\frac{1}{2n_i} \ln\left(\frac{2n}{\mu}\right)} \) with probability \( 1 - \frac{\mu}{n} \)
- Substituting \( l = \frac{2}{(\ln^{-1}(\Delta))^2} \ln\left(\frac{2n}{\mu}\right) \), \( \hat{q}^+_i - q_i \leq h^{-1}(\Delta) \) with probability \( 1 - \frac{\mu}{n} \)
- By union bound, \( \hat{q}^+ - q \leq h^{-1}(\Delta) \) with probability \( 1 - \mu \)
- By bounded smoothness property, \( |f_S(\hat{q}^+) - f_S(q)| \leq h(h^{-1}(\Delta)) \leq \Delta \) with probability \( 1 - \mu \)
- Similarly, \( \forall S \subseteq \mathcal{N}, |f_S(\hat{q}^-) - f_S(q)| \leq \Delta \) with probability \( 1 - \mu \)

Thus, \( f_S(q) - \Delta \leq f_S(\hat{q}^+) \leq f_S(q) + \Delta \) and \( f_S(\hat{q}^-) - \Delta \leq f_S^*(q) \leq f_S^*(\hat{q}^+) + \Delta \)

Thus, \( f_S(q) > \alpha \Rightarrow f_S(\hat{q}^+) > \alpha \) (\( \because f_S(q) > \alpha + \Delta \), \( \Delta \)-separated property)

And, \( f_S^*(\hat{q}^-) < \alpha \) (\( \because f_S^*(q) < \alpha - \Delta \), \( \Delta \)-separated property)

\[ \square \]

**Lemma 4.3** If a non-optimal set \( S^t \) is selected for task \( t \) then there exists a worker \( i \in S^t \) such that \( n_{i,t} \leq \frac{2}{(\ln^{-1}(\Delta))^2} \ln\left(\frac{2n}{\mu}\right) \) with probability \( 1 - \mu \)

**Proof:** A non-optimal subset \( S^t \) could be selected in two ways:

- \( f_{S^t}(\hat{q}^+) < \alpha \) but \( f_{S^t}(q) > \alpha \)
- \( f_{S^t}(\hat{q}^-) > \alpha \)

From Lemma 4.2, if \( n_{i,t} \geq \frac{2}{(\ln^{-1}(\Delta))^2} \ln\left(\frac{2n}{\mu}\right) \) \( \forall i \in S^t \), then both the conditions are violated and thus non-optimal subset is not selected.

\[ \square \]

**Theorem 4.2** The number of non-optimal rounds by the CCB-NS algorithm is bounded by \( \frac{2n}{(\ln^{-1}(\Delta))^2} \ln\left(\frac{2n}{\mu}\right) \) with probability \( 1 - \mu \).

**Proof:**

- Lemma 4.1 shows that CCB-NS exploitation rounds are optimal rounds
- A new parameter \( u_{i,t} \) is associated with each worker. Whenever a set \( S^t \) is selected then, \( u_{i,t} = u_{i,t} + 1 \) s.t \( i \in S^t \) and \( i = \arg\min_{j \in S^t} u_{j,t} \)
- Every time a non-optimal subset \( S^t \) is selected, \( u_{i,t} \) of only one worker is updated with lowest value of \( u_{i,t} \) so far, such that \( i \in S^t \). Thus, \( u_{i,t} \leq n_{i,t} \) \( \forall i \in \mathcal{N} \) \( \forall t \in \{1, \ldots, T\} \)
- Thus, from Lemma 4.3, the number of exploration rounds is bounded by \( \frac{2n}{(\ln^{-1}(\Delta))^2} \ln\left(\frac{2n}{\mu}\right) \) with probability \( 1 - \mu \).

Hence the theorem follows.

\[ \square \]

We now present the strategic version in the next section and see how the CCB-NS algorithm can be modified in a simple way so as to satisfy desirable game theoretic properties. We start by first defining the model in game theoretic setting. We then present the CCB-S algorithm and derive the regret bounds. Finally, we show that the allocation rule provided by the CCB-S algorithm is ex-post monotone and hence results in ex-post incentive compatible and ex-post individually rational mechanism.
5 Strategic Version

Here, instead of knowing the cost of each worker beforehand, we ask the costs and design the payment scheme so that it is in the best interest of every agent to report her true cost. We start by defining the valuations and utility functions of the requester and the workers. Then we show that the algorithm CCB-S presented above is ex-post monotone in terms of worker’s cost and thus, the approach from [5] can be used to design the payments scheme so as to form an ex-post incentive compatible and ex-post individually rational mechanism in our setting.

5.1 The Model

Denote the true cost of a worker $i$ by $c_i$ and the reported cost by $\hat{c}_i$. Valuation of a worker $i$ is given by $v_i = -c_i$. For convenience, we will denote the requester as agent 0 and valuation of requester on allocating task to the set of workers $S$ is given by:

$$v_0(S) = \begin{cases} R, & \text{if } f_S(q) < \alpha \\ -\hat{C}, & \text{otherwise} \end{cases}$$

where, $f_S(q)$ is the error probability function that satisfies the properties of monotonicity and bounded smoothness and $1 - \alpha$ is the target accuracy. Here, $R$ denotes the reward that the requester gets for satisfying the constraint and $\hat{C}$ is the cost he incurs if the constraint in Equation 2 is not satisfied. Note that the requester is not considered to be strategic. Social welfare $W(S)$ is given by the sum of valuation of all the agents if set $S$ is selected:

$$W(S) = \begin{cases} R - \sum_{i \in S} c_i, & \text{if } f_S(q) < \alpha \\ -\hat{C} - \sum_{i \in S} c_i, & \text{otherwise} \end{cases}$$

The focus of this work is to impose the constraint as a hard constraint. And, thus $R$ and $\hat{C}$ are designed in such a way that for all tasks $t$ completing task with given accuracy yields more social welfare compared to the case where it fails to maintain the given accuracy. Thus,

$$R - \sum_{i \in S} c_i > -\hat{C} \forall S \subseteq N$$

A mechanism $\mathcal{M}$ is denoted by the pair $(A, P)$, where $A$ is the allocation function and $P$ is the payment function which depends on the reported cost profile $\hat{c}$ and the learnt quality profile $\hat{q}$. We work in a quasi-linear setting where utility of every agent is given by:

$$u_i(c_i, \hat{c}) = -c_i.A_i(\hat{c}, \hat{q}) + P_i(\hat{c}, \hat{q})$$

Since there is randomness involved in our algorithm due to learnt qualities, let us first define every possible random seed. The random variables are the noisy labels that are provided by the workers and can affect the learnt qualities and thus the allocation rule (or the length of the exploration phase).

**Definition 5.1 (Success Realization)** A success realization is a matrix $\rho \in \{0, 1, -1\}^{n \times T}$.

$$\rho_{it} = \begin{cases} 1 & \text{if } \hat{y}_i^t = y_t \\ 0 & \text{if } \hat{y}_i^t \neq y_t \\ -1 & \text{if agent } i \text{ is not selected in the } t^{th}\text{round} \end{cases}$$

We now review some of the desirable properties that mechanism $\mathcal{M}$ should satisfy:
Definition 5.2 (Truthfulness) A mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ is said to be truthful if reporting true valuations is a dominant strategy for all the workers. That is, $\forall i \in \mathcal{N}$,

$$-c_i A_i(c_i, \hat{c}_{-i}) + P_i(c_i, \hat{c}_{-i}) \geq -c_i A_i(\hat{c}_i, \hat{c}_{-i}) + P_i(\hat{c}_i, \hat{c}_{-i}) \quad \forall c_i \in [0, 1] \text{ and } \hat{c}_{-i} \in [0, 1]^{n-1}$$

Definition 5.3 (Interim Individual Rationality) A mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ is said to be individually rational for a worker if participating in the mechanism always gives him positive utility. That is, $\forall i \in \mathcal{N}$,

$$-\hat{c}_i A_i(\hat{c}_i, c_{-i}) + P_i(\hat{c}_i, c_{-i}) \geq 0 \quad \forall \hat{c}_i \in [0, 1] \text{ and } c_{-i} \in [0, 1]^{n-1}$$

An important characterization for truthful mechanisms provided by Myerson [23] states that for a mechanism to be truthful, the allocation rule should be monotone in terms of reported bids by the players. Babaioff et. al. [5] provides a generic transformation that takes any monotone allocation rule and outputs a mechanism which is truthful and individually rational. We can use this generic transformation to design the mechanism in our setting. We first provide the definition of monotonicity in our setting.

Definition 5.4 (Monotonicity of Allocation Rule) An allocation rule $\mathcal{A}$ is monotone if for every worker $i$, and for every fixed $\hat{c}_{-i} \in [0, 1]^{n-1}$,

$$\hat{c}_i \leq \hat{c}_i^+ \Rightarrow A_i(\hat{c}_i, \hat{c}_{-i}) \geq A_i(\hat{c}_i^+, \hat{c}_{-i})$$

where $A_i(\hat{c}_i, \hat{c}_{-i})$ is number of tasks given to the $i^{th}$ worker with bids $\hat{c}_i$ and $\hat{c}_{-i}$.

Since there is randomization in our mechanism (due to learned qualities), we also define relevant weaker notions of truthfulness, individual rationality and monotonicity. Note that, the allocation and payment rule will also depend on success realizations.

Definition 5.5 (Ex-Post Incentive Compatibility) We say that a mechanism is ex-post incentive compatible if all the bidders are truthful for every success realization irrespective of the bids of other workers,

$$-c_i A_i(c_i, \hat{c}_{-i}, \rho) + P_i(c_i, \hat{c}_{-i}, \rho) \geq -c_i A_i(\hat{c}_i, \hat{c}_{-i}, \rho) + P_i(\hat{c}_i, \hat{c}_{-i}, \rho)$$

$\forall \hat{c}_i \in [0, 1], \hat{c}_{-i} \in [0, 1]^{n-1}, \rho \in \{0, 1, -1\}^{n \times T}$

Definition 5.6 (Universally Ex-Post Individual Rationality) We say that a mechanism is universally ex-post individual rational if for every success realization, truth telling does not give negative utility to any player corresponding to any bids of other players,

$$-\hat{c}_i A_i(\hat{c}_i, c_{-i}, \rho) + P_i(\hat{c}_i, c_{-i}, \rho) \geq 0 \quad \forall \hat{c}_i \in [0, 1], c_{-i} \in [0, 1]^{n-1}, \rho \in \{0, 1, -1\}^{n \times T}$$

Definition 5.7 (Ex-Post Monotone Allocation Rule) If the allocation rule is monotone with respect to every success realization then we say that it is ex-post monotone.

Based on the above preliminaries for truthful implementation of a MAB algorithm, we now present the strategic version of CCB-NS algorithm which we call, the CCB-S algorithm.

5.2 The CCB-S Algorithm

As we have seen in previous section, we need monotonicity of an allocation rule for truthful implementation. In CCB-NS, if selected set $S^t$ in step 6 does not satisfy constraint with $\hat{q}^-$, in step 8, we add agents to satisfy constraint. This step does not consider strategic costs and
thus leads to violation of monotonicity. Hence, we design a new algorithm which achieves the necessary monotonicity.

In order to ensure truthfulness, we modify the CCB-NS algorithm to select a complete set of workers if the constraint is not satisfied with respect to the lower confidence bound. We then show that the allocation rule given by the algorithm is ex-post monotone and thus one can design a payment rule so as to ensure the truthfulness. Thus, we can apply results from [5] to achieve an ex-post incentive compatible and ex-post individual rational mechanism. Before going to the formal analysis of CCB-S algorithm, we first formally present an important result in [5] in our setting:

**Theorem 5.1** [5] Consider the stochastic MAB mechanism design problem. Let the parameter provided to the transformation.

There exists a transformation in our setting:

Note that the randomness is coming due to the qualities that are being learned. The results from [5] says that in order to design an ex-post incentive compatible and ex-post individual

### Algorithm 2: CCB-S Algorithm

**Input:** Parameter $\alpha$, number of tasks $T$, set of workers $N$, confidence level $\mu$

**Output:** Labeler selection set $S^t$, Label $\hat{y}_t$ for task $t$

1. $\forall i \in N$, $\hat{q}^+_i = 1$, $\hat{q}^-_i = 0.5$, $c_{i,1} = 0$ \hspace{1em} Initialize UCB and LCB on qualities
2. $S^1 = N$, observe $\hat{y}(S^1)$ and $\hat{y}_1 = \text{AGGREGATE}(S^1)$ \hspace{1em} Select all workers initially
3. Observe true label $y_1$
4. $\forall i \in N$, $n_{i,1} = 1$, $c_{i,1} = 1$ if $\hat{y}_i = y_1$ and $\hat{q}_i = c_{i,1}/n_{i,1}$
5. for $t = 2$ to $T$ do
6. \hspace{1em} Let $S^t = \text{arg min}_{S \subseteq N} \sum_{i \in S} c_i$ s.t. $f_S(\hat{q}^+) < \alpha$
7. \hspace{1em} if $f_S^1(\hat{q}^-) > \alpha$ then
8. \hspace{2em} $S^t = N$, $\hat{y}_t = \text{AGGREGATE}(S^t)$ \hspace{1em} Explore
9. \hspace{2em} Observe true label $y_t$
10. \hspace{1em} $\forall i \in S^t$, $n_{i,t} = n_{i,t} + 1$, $c_{i,t} = c_{i,t} + 1$ if $\hat{y}_i = y_t$, $\hat{q}_i = c_{i,t}/n_{i,t}$,
    
    $\hat{q}^+_i = \hat{q}_i + \sqrt{\frac{1}{2n_{i,t}} \ln \left( \frac{2n}{\mu} \right)}$, $\hat{q}^-_i = \hat{q}_i - \sqrt{\frac{1}{2n_{i,t}} \ln \left( \frac{2n}{\mu} \right)}$
11. else
12. \hspace{1em} $t^* = t$
13. \hspace{1em} $\hat{y}_t = \text{AGGREGATE}(S^t)$
14. \hspace{1em} Break \hspace{1em} Goto Step 15
15. for $t = t^* + 1$ to $T$ do
16. \hspace{1em} $S^t = S^{t^*}$, $\hat{y}_t = \text{AGGREGATE}(S^t)$ \hspace{1em} Exploit

The allocation induced by the CCB-S algorithm is denoted by $A^{CCB-S}$. A mechanism $M^{CCB-S} = \{A^{CCB-S}, P^{CCB-S}\}$ is called a CCB-S mechanism when the allocation rule $A^{CCB-S}$ is the transformation of $A^{CCB-S}$ given by [5] and the payment rule is given in [4]. We now present a game theoretic analysis of the mechanism $M^{CCB-S}$.

### 5.3 Analysis of $M^{CCB-S}$

Note that the randomness is coming due to the qualities that are being learned. The results from [5] says that in order to design an ex-post incentive compatible and ex-post individual
rational mechanism, it is enough to design an ex-post monotone allocation rule. Thus, we will show that our algorithm achieves ex-post monotonicity and hence using tools from [5], we can achieve an ex-post truthful and ex-post individually rational mechanism (Theorem 5.1).

Since the algorithm turns to be exploration separated, one natural way to ensure truthfulness is to apply the classical VCG mechanism. We cannot apply VCG payment scheme in this algorithm as computing VCG payments requires computation of an allocation rule in absence of worker \( i \) which cannot be determined by the algorithm since learning stops after computing the optimal set.

**Theorem 5.2** Allocation rule given by the CCB-S algorithm \( \mathcal{A}^{CCB-S} \) is ex-post monotone.

**Proof:** For notation brevity, let us denote \( \mathcal{A}^{CCB-S} \) by \( A \). In order to prove monotonicity, we need to prove the following:

\[
A_i^t(\hat{c}_i, c_{-i}) \leq A_i^t(c_i, c_{-i})
\]

\[\forall i \in N, \forall t \in \{1, 2, \ldots, T\}, \forall \hat{c}_i \geq c_i \]

Since task \( t = 1 \) is given to all the workers irrespective of their bids, we have \( A_i^1(\hat{c}_i, c_{-i}) = A_i^1(c_i, c_{-i}) = 1 \forall j \). Let \( t \) be the largest time step such that, \( \forall j, A_j^{t-1}(\hat{c}_i, c_{-i}) = A_j^{t-1}(c_i, c_{-i}) = t - 1 \) (Exploration round with \( \hat{c}_i \) and \( c_i \)). And \( \exists i \) such that,

\[A_i^t(\hat{c}_i, c_{-i}) \neq A_i^t(c_i, c_{-i})\]

Since other costs and quality estimates are the same, this can happen only when in one case worker \( i \) is selected, while in other case worker \( i \) is not selected. Let the two sets selected with \( c_i \) and \( \hat{c}_i \) be \( S(c_i) \) and \( S(\hat{c}_i) \) respectively. Since the optimization problem involves cost minimization and quality updates are the same, we have,

\[A_i^t(\hat{c}_i, c_{-i}) = t - 1 \quad \text{which implies} \quad i \notin S(\hat{c}_i)\]

\[A_i^t(c_i, c_{-i}) = t \quad \text{which implies} \quad i \in S(c_i)\]

Since \( i \notin S(\hat{c}_i) \), selected set \( S(\hat{c}_i) \) satisfies the lower confidence bound too (exploitation round with bid \( \hat{c}_i \)) and thus for the rest of the tasks, only \( S(\hat{c}_i) \) is selected and thus we have,

\[A_i^t(\hat{c}_i, c_{-i}) \leq A_i^t(c_i, c_{-i})\]

\[\Box\]

We denote the mechanism \( \mathcal{M}^{CCB-S} = (\mathcal{A}^{CCB-S}, \mathcal{P}^{CCB-S}) \). As given by Theorem 5.1, \( \mathcal{P}^{CCB-S} \) can be derived by applying the transformation given in [5]. Thus, we obtain the following corollary:

**Corollary 5.3** CCB-S algorithm produces an ex-post incentive compatible and ex-post individually rational mechanism.

One can apply transformation presented in [5] which takes any ex-post monotone allocation rule as input and outputs a randomized mechanism which is ex-post incentive compatible and universally ex-post individually rational.

**Regret Analysis**

The proposed algorithm is adaptive exploration separated and the number of exploration steps is decided based on how learning progresses. We will show that \( t^* \) returned by Step 11 of algorithm is bounded by \( \frac{2}{(n - \tau(S))^2} \ln(\frac{2n}{\mu}) \). In Lemma 5.1 we prove that after \( l = \frac{2}{(n - \tau(S))^2} \ln(\frac{2n}{\mu}) \), there is no set \( S \) which satisfies the constraint with respect to upper confidence bound and its cost is less then the optimal cost. Moreover, after \( l = \frac{2}{(n - \tau(S))^2} \ln(\frac{2n}{\mu}) \), we have \( f_S(\hat{q}^*) < \alpha \) with probability \( 1 - \mu \).
Lemma 5.1 After \( l = \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu}) \) number of uniform exploration rounds,

1. for all sets \( S \neq S^* \), \( f_S(q) > \alpha \Rightarrow f_S(\hat{q}^+) > \alpha \) with probability \( 1 - \mu \)

2. \( f_{S^*}(\hat{q}^-) < \alpha \) with probability \( 1 - \mu \)

Proof:

- By Hoeffding’s inequality, \( \hat{q}^+ - q_i \leq 2 \sqrt{\frac{1}{2n_i} \ln(\frac{2}{\mu})} \leq 2 \sqrt{\frac{1}{2} \ln(\frac{2n}{\mu})} \) with probability \( 1 - \frac{\mu}{n} \)
- Substituting \( l = \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu}) \), \( \hat{q}^+ - q_i \leq h^{-1}(\Delta) \) with probability \( 1 - \frac{\mu}{n} \)
- By union bound, \( \hat{q}^+ - q \leq h^{-1}(\Delta) \) with probability \( 1 - \mu \)
- By bounded smoothness property, \( |f_S(\hat{q}^+) - f_S(q)| \leq h(h^{-1}(\Delta)) \leq \Delta \) with probability \( 1 - \mu \).
- Similarly, \( \forall S \subseteq \mathcal{N}, |f_S(\hat{q}^-) - f_S(q)| \leq \Delta \) with probability \( 1 - \mu \)

Thus, \( f_S(q) - \Delta \leq f_S(\hat{q}^+) \leq f_S(q) + \Delta \) and \( f_{S^*}(\hat{q}^-) - \Delta \leq f_{S^*}(q) \leq f_{S^*}(\hat{q}^-) + \Delta \)

Thus, \( f_S(q) > \alpha \Rightarrow f_S(\hat{q}^+) > \alpha \) (\( f_S(q) > \alpha + \Delta, \Delta \)-separated property)

And, \( f_{S^*}(\hat{q}^-) < \alpha \) (\( f_{S^*}(q) < \alpha - \Delta, \Delta \)-separated property)

□

As a result of lemmata 4.1 and 5.1 we have the following theorem which gives us the bound on number of non-optimal rounds:

Theorem 5.4 Number of non-optimal rounds by the CCB-S algorithm is bounded by \( \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu}) \) with probability \( 1 - \mu \).

Proof:

- From Lemma 4.1 the CCB-S exploitation rounds are optimal rounds
- From Lemma 5.1 the number of exploration rounds is bounded by \( \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu}) \) with probability \( 1 - \mu \).

Hence the theorem follows.

□

Remark: The algorithm CCB-S turns out to be an exploration separated algorithm where the number of exploration steps is adaptive unlike the algorithms presented in [13, 17] and bounds on the number of exploration steps depend on the parameters \( \Delta \) and \( \mu \).

Thus, we have the following Corollary that follows from Theorem 5.4:

Corollary 5.5 The total expected regret is bounded by

\[
(1 - \mu) \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu}) C(\mathcal{N}) + \mu \tilde{C} T
\]

where \( \tilde{C} \) is cost incurred by the requester if constraint is not satisfied.

This dependence on \( \Delta \) arises, because we are trying to satisfy the constraint with respect to the lower confidence bound.
6 Discussion

6.1 High Cost of exploration

Selecting all workers in exploration step if constraint is not satisfied with lower confidence bound, might result in very high cost of exploration if there are huge number of workers. All the workers are required to be explored to ensure truthfulness however, if costs are public knowledge or if we assume that workers bid their true cost, then instead of selecting all the workers, one may add random minimal workers from set $N \setminus S^t$ in each round till the constraint with lower confidence bound is satisfied. The algorithm is presented in Algorithm 2 which is a slight modification of Algorithm 1 where instead of selecting all the workers in exploration rounds, minimal set satisfying the constraint with lower confidence bound is selected. This might result in increase in number of bad rounds by at most a factor of $n$ since learning will be slow but then each bad round would incur less cost as compared to the one in strategic case since much lesser number of workers needs to be selected if qualities are high enough.

6.2 Comparison with CUCB algorithm [11]

Though our setting deals with combinatorial arms similar to [11], AAB framework is different as hard constraint needs to be satisfied at every round as oppose to traditional combinatorial MAB setting. As noted earlier, monotonicity and bounded smoothness properties are considered by Chen et al. for the reward function. Since the learning parameters are coming in the constraint in our setting, we have considered these assumptions on the constraint function. The CUCB achieves regret which is proportional to $\ln T$. In particular, the regret of CUCB algorithm is given by $O\left(\frac{\ln T}{(h^{-1}(\Delta_{min}))^2}\Delta_{max}\right)$ where

$$\Delta_{min} = \min_i \{r_{S^*}(q) - \max_{S \in S_B} r_S(q)|S \in S_B, i \in S\}$$

$$\Delta_{max} = \max_i \{r_{S^*}(q) - \min_{S \in S_B} r_S(q)|S \in S_B, i \in S\}$$

where $r$ is the reward function and $S_B$ is the set of all non-optimal subsets. $h$ is the monotone continuous function, such that,

$$\max_i |q_i - q'_i| < \delta \implies |r_S(q) - r_S(q')| < h(\delta)$$

As can be seen, again $h$ function on the constraint in our setting is similar to the $h$ function in the reward function. Thus, our regret expression looks similar to the one given in [11].

7 Summary and Future Work

Motivated by the need for a mechanism where worker qualities are not known a priori and their costs are private information, we considered the problem of selecting an optimal subset of workers to work on the same task so that the outcome obtained from aggregating labels from the selected workers attains a minimum accuracy level. We proposed a novel framework Assured Accuracy Bandit (AAB) in this setting and developed an algorithm Strategic Constrained Confidence Bound (CCB-S) for the same which also leads to an ex-post incentive compatible and ex-post individually rational mechanism. The mechanism presented is unique in that it learns qualities of strategic workers who have costs as private information. We have provided bounds on exploration steps for a given problem instance that depends on the target accuracy and true qualities.

The optimization problem involved is a hard problem to solve. The algorithm works well when the number of workers is small enough and qualities are high enough. For uniform costs,
one can solve the optimization problem greedily and since it is not strategic, instead of selecting all workers, one can select a minimal set to satisfy the constraint in the exploration phase.

The algorithm is currently exploration separated in order to achieve ex-post monotonicity. One can investigate existence of other algorithms satisfying desirable mechanism properties with lower regret. It would be interesting to provide a lower bound on regret in this setting. Currently, the proposed monotonic constraint is stiffer than the desirable constraint. It would be nice to see if the constraint can be relaxed closer to the actual one. Currently, we are using majority voting rule, one can also use different other rules to aggregate crowd answers like weighted majority voting. The challenge in such a setting is to come up with such rules taking into account the fact that qualities are not known a priori.

References

[1] Ittai Abraham, Omar Alonso, Vasilis Kandylas, and Aleksandrs Slivkins. Adaptive crowdsourcing algorithms for the bandit survey problem. In Shai Shalev-Shwartz and Ingo Steinwart, editors, Conference On Learning Theory, volume 30 of JMLR Proceedings, pages 882–910. JMLR.org, 2013.

[2] Shipra Agrawal and Nikhil R. Devanur. Bandits with concave rewards and convex knapsacks. In Fifteenth ACM Conference on Economics and Computation, To appear, 2014.

[3] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. Journal of Machine Learning, 47(2-3):235–256, May 2002.

[4] Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. In Thirteenth ACM Conference on Electronic Commerce, pages 74–91. ACM, 2012.

[5] Moshe Babaioff, Robert D. Kleinberg, and Aleksandrs Slivkins. Truthful mechanisms with implicit payment computation. In Eleventh ACM Conference on Electronic Commerce, pages 43–52. ACM, 2010.

[6] Moshe Babaioff, Yogeshwer Sharma, and Aleksandrs Slivkins. Characterizing truthful multi-armed bandit mechanisms: extended abstract. In Tenth ACM Conference on Electronic Commerce, pages 79–88. ACM, 2009.

[7] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: posted price mechanisms for online procurement. In Thirteenth ACM Conference on Electronic Commerce, pages 128–145. ACM, 2012.

[8] Satyanath Bhat, Swaprava Nath, Onno Xoeter, Sujit Gujar, Yadati Narahari, and Chris Dance. A mechanism to optimally balance cost and quality of labeling tasks outsourced to strategic agents. In Thirteenth International Conference on Autonomous Agents and Multiagent Systems, pages 917–924, 2014.

[9] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Foundations and Trends in Machine Learning, 5(1):1–122, 2012.

[10] Ruggiero Cavallo and Shaili Jain. Winner-take-all crowdsourcing contests with stochastic production. In Proceedings of the first AAAI Conference on Human Computation and Crowdsourcing, 2013.
[11] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, volume 28, pages 151–159, 2013.

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

[13] Nikhil R. Devanur and Sham M. Kakade. The price of truthfulness for pay-per-click auctions. In *Tenth ACM Conference on Electronic Commerce*, pages 99–106, 2009.

[14] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. Multi-armed bandit with budget constraint and variable costs. In *Twenty Seventh Conference on Artificial Intelligence*. AAAI Press, 2013.

[15] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning*, 7:1079–1105, 2006.

[16] Dinesh Garg, Sourangshu Bhattacharya, S. Sundararajan, and Shirish Krishnaj Shevade. Mechanism design for cost optimal PAC learning in the presence of strategic noisy annotators. In *Twenty Eighth Conference on Uncertainty in Artificial Intelligence*, pages 275–285, 2012.

[17] Nicola Gatti, Alessandro Lazaric, and Francesco Trovò. A truthful learning mechanism for contextual multi-slot sponsored search auctions with externalities. In *Thirteenth ACM Conference on Electronic Commerce*, pages 605–622, 2012.

[18] Shweta Jain, Sujit Gujar, Onno Xoeter, and Y. Narahari. A quality assuring multi-armed bandit crowdsourcing mechanism with incentive compatible learning. In *Thirteenth International Conference on Autonomous Agents and Multiagent Systems*, pages 1609–1610, 2014.

[19] Chien ju Ho, Shahin Jabbari, and Jennifer W. Vaughan. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning*, volume 28, pages 534–542, 2013.

[20] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *International Conference on Machine Learning*, 2010.

[21] Oh Sewoong Karger David R. and Shah Devavrat. Budget-optimal crowdsourcing using low-rank matrix approximations. In *49th Annual Conference on Communication, Control, and Computing (Allerton)*, pages 284–291, 2011.

[22] Hongwei Li, Bin Yu, and Dengyong Zhou. Error rate bounds in crowdsourcing models. In *arXiv preprint arXiv:1307.2674*, 2013.

[23] Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):pp. 58–73, 1981.

[24] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *The Journal of Machine Learning Research*, 11:1297–1322, 2010.

[25] Akash Das Sharma, Sujit Gujar, and Y. Narahari. Truthful multi-armed bandit mechanisms for multi-slot sponsored search auctions. *Current Science*, Vol. 103 Issue 9, 2012.
[26] Yaron Singer and Manas Mittal. Pricing mechanisms for crowdsourcing markets. In *Twenty Second Internation World Wide Web Conference*, pages 1157–1166, 2013.

[27] Adish Singla and Andreas Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Twenty Second International World Wide Web Conference*, pages 1167–1178, 2013.

[28] Long Tran-Thanh, Archie C. Chapman, Alex Rogers, and Nicholas R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *Twenty-Sixth Conference on Artificial Intelligence (AAAI 2012)*, 2012.

[29] Paolo Viappiani, Sandra Zilles, HowardJ. Hamilton, and Craig Boutilier. Learning complex concepts using crowdsourcing: A Bayesian Approach. In *Algorithmic Decision Theory*, volume 6992, pages 277–291. 2011.

[30] Jens Witkowski, Yoram Bachrach, Peter Key, and David C. Parkes. Dwelling on the Negative: Incentivizing Effort in Peer Prediction. In *Proceedings of the first AAAI Conference on Human Computation and Crowdsourcing (HCOMP’13)*, 2013.