Multi-Armed Bandit Algorithms for Crowdsourcing Systems with Online Estimation of Workers' Ability

Anshuka Rangi
Department of Electrical and Computer Engineering,
University of California, San Diego
California, USA
arangi@ucsd.edu

Massimo Franceschetti
Department of Electrical and Computer Engineering,
University of California, San Diego
California, USA
massimo@ece.ucsd.edu

ABSTRACT

Crowdsourcing systems have become a valuable solution for various organizations to outsource work on a temporary basis. Quality assurance in these systems remains a key issue due to the distributed setup of the crowdsourcing platforms and the absence of a priori information about the workers. Our work develops a notion of Limited-information Crowdsourcing Systems (LCS), where the task master can assign the work based on some knowledge of the workers' ability acquired over time. The key challenges in this new setup are determining an efficient workers' selection policy and estimating the abilities of the workers. To address the first challenge, we reduce the problem to an arm-limited, budget limited, multi-armed bandit (MAB) set-up, and use the simplified bounded KUBE (B-KUBE) algorithm as a solution. This algorithm has previously only been experimentally evaluated, and we provide provable performance guarantees, showing that it is order optimal, namely the expected regret of B-KUBE is $O(\log(B))$ where B is the total budget of the task master. The second challenge is solved by formalizing the notion of workers' ability mathematically, and proposing a strategy for its estimation. We experimentally evaluate B-KUBE in conjunction with this strategy, showing that it outperforms other state-of- the-art MAB algorithms when applied in the same setting.

KEYWORDS

Multi-Armed Bandits; Crowdsourcing Systems; Bounded Knapsack Problem.

ACM Reference Format:
Anshuka Rangi and Massimo Franceschetti. 2018. Multi-Armed Bandit Algorithms for Crowdsourcing Systems with Online Estimation of Workers' Ability. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 8 pages.

1 INTRODUCTION

Crowdsourcing systems (CS) have emerged as a valuable tool for several organizations to outsource a variety of tasks to a population of diverse workers at low cost. Some of the key players in the crowdsourcing market include for example *Amazon Mechanical Turk, Upwork, Freelancer* and *uTest*. In these systems, guaranteeing the quality of the work remains a key challenge, due to the limited a priori information about the ability of the workers. Thus, there is interest in developing automated methods for the collection and aggregation of information from the workers, incentive schemes to hire expert workers, and schemes for determining the quality of the tasks being done.

CS research has mostly focused on distributed methods where there is very limited interaction between workers and task master. The interaction is typically limited to the assignment of a gold-set of tasks to evaluate workers' performance prior to the assignment of the actual set of tasks, and does not provide a way to continuously monitor the quality of the work in real time. Dishonest workers can perform well on a gold-set of tasks and, not being evaluated on-line on a competitive basis, underperform during the actual working phase. Alternatively, the gold set can be mixed with all the assigned tasks in a way that the workers cannot distinguish between them. This is helpful to detect underperforming workers, but it wastes resources, and does not ensure continuous monitoring of the quality of the work.

In this paper, we develop a notion of *Limited-information Crowdsourcing Systems* (LCS) that is desirable from both the task master and workers perspective. In LCS, workers express their interest in doing the tasks, quote their charges per task, and provide an upper limit on the number of tasks they are willing to perform. The tasks can then be assigned in burst or one-by-one to the workers, as long as the workers' constraints are satisfied. Given these constraints, unlike traditional CS, the workers do not need to be assigned all of their tasks at the same time. The workers' selection policy is not limited to be of the form "take-it" or "leave-it," but it can include workers who are still available after having completed a certain number of tasks, and that may be assigned additional tasks at a later time. This eliminates the requirement of having gold-set of tasks, and allows the task master to continuously monitor the quality of the work and assign tasks based on the estimated workers' ability, thus creating a competitive environment. Additionally, the workers are incentivized to perform tasks satisfactorily in order to maximize their earnings, while satisfying their load constraint.

This new formulation also poses new challenges. In our setting, the workers' selection algorithm needs to balance an exploration-exploitation trade-off, since the workers' ability is initially unknown to the task master and is learned on-line. This trade-off is not considered in traditional CS due to the limited interaction between the workers and the task master, but it is a classic one in the field of Multi-Armed Bandits (MAB) [18]. This is a class of problems dealing with decision making under uncertainty, where the actions have rewards that have to be learned through observations. Thus, the main challenge in LCS is to determine an efficient workers' selection scheme and to estimate of the abilities of the workers. To exploit the similarity of LCS with MAB, we reformulate the LCS

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

problem in terms of a Bounded Knapsack Problem (BKP) that is equivalent to an arm-limited, budget-limited MAB. Given a strategy to estimate the workers' ability in real time, we use the B-KUBE algorithm developed in [23] for workers' selection. This algorithm has previously only been evaluated experimentally, and we provide provable performance guarantees, showing that its expected regret is $O(\log B)$, where B is the maximum available budget. Since it has been shown in [4] that the expected regret for any algorithm is at least $\Omega(\log B)$, our results imply that B-KUBE is order optimal. Thus, we close the gap in the literature of arm-limited, budget-limited MAB by providing the first order optimal bounds of an algorithm in the current MAB setup. We then formalize the notion of workers' ability and propose an online strategy to estimate it. We also experimentally evaluate B-KUBE in conjunction with our strategy for estimating the workers' ability, showing that it outperforms other state-of-the-art MAB algorithms applied in the same setting. Thus, the contributions of the work are two fold: providing an optimal scheme for a MAB setup and using it in a crowdsourcing setting conjunction with an estimation scheme.

The organization of the paper is as follows: Section 2 describes the problem formulation; Section 3 discusses related work; Section 4 describes usage of B-KUBE for workers' selection and gives its performance guarantees; section 5 describes a strategy for estimating the workers' performance in real time; section 6 provides experimental evaluation of of B-KUBE in conjunction with this strategy; section 7 concludes the work.

2 PROBLEM FORMULATION

We consider a labeling task in LCS, but this formulation can be easily modified to accommodate a different type of work. We assume the task master has a budget B and needs to label data with one of L labels. There are K workers interested in performing the labeling tasks. For every $k \in [K]$, the number of evaluations a worker can perform is limited by M_k and the cost of each evaluation is c_k. The objective of the task master is to minimize the average classification error

$$\epsilon = \frac{1}{T}\sum_i \mathbb{P}(\hat{l}_i \neq l_i^*) \tag{1}$$

where $\hat{l}_i$ and l_i^* are the predicted label and true label of the task i respectively, and T is the total number of labeling tasks. This is a common measure of performance considered in crowdsourcing systems works [14, 15, 17]. Thus, letting x_k be the number of evaluations assigned to each worker, we define the LCS problem as follows

$$\begin{aligned} &\min \epsilon \text{ s.t.}\\ &\sum_k x_k c_k \leq B,\\ &\forall k \in [K] : 0 \leq x_k \leq M_k,\\ &\text{and } x_k \text{ is an integer.}\end{aligned} \tag{2}$$

We now reformulate the problem in (2) as a Bounded Knapsack problem (BKP). Assume that the measure of a worker's performance is given by a value contribution v_k. This value contribution is a measure of information contributed by a worker to the system after each evaluation. Minimizing ϵ in the LCS problem is then analogous to maximizing the aggregated value contributions in the following BKP

$$\begin{aligned} &\max_{\{x_k\}} \sum_k x_k v_k \text{ s.t.}\\ &\sum_k x_k c_k \leq B,\\ &\forall k \in [K] : 0 \leq x_k \leq M_k,\\ &\text{and } x_k \text{ is an integer.}\end{aligned} \tag{3}$$

The key benefit of the reformulation to BKP is that it provides an insight on the optimal aggregation of two different attributes, cost and value contribution, of the workers. Despite this equivalent formulation, the original LCS problem cannot be solved using standard BKP techniques. The value contributions typically are assumed to be known in BKP [16], while they need to be estimated in our setting. Nevertheless, the problem in (3) is also equivalent to an arm-limited, budget-limited stochastic MAB problem, whose expected rewards correspond to the unknown value contributions.

In a stochastic MAB problem, there are K arms of a single "bandit" machine. Pulling of each arm delivers a reward that is independently drawn from an unknown distribution. An agent chooses to pull arms with the goal of maximizing the expected sum of the rewards received over a sequence of pulls.

We consider a popular stochastic model, from the literature of CS, for modeling the workers' responses. In this model, a worker k can be assigned a task multiple times and the correct label is predicted each time with probability p_k independent of the past responses of the worker about the task [1, 11, 14, 15, 17, 23, 26]. Given a task i, for all workers $k \in [K]$, we assume that the probability of predicting any incorrect label is the same for all labels independent of the task i and true label l_i^*, namely for all $\hat{l}_{i,k} \in [L]$ we have $P(\hat{l}_{i,k} \neq l_i^*) = (1 - p_k)/L - 1$, where $\hat{l}_{i,k}$ is the predicted label of task i by the worker k. We also assume that the value contribution of a worker remains the same irrespective of the true label, namely for all $i \in [T]$ and $l_i^* \in [L]$ we have $v_k(l_i^*) = v_k$. These assumptions are only made for ease of presentation of our estimation strategy for value contributions and all the theoretical results provided in the paper do not rely on them.

The workers in LCS are equivalent to arms in MAB, and the task master plays the same role as the agent in MAB. The value contributions of the workers are analogous to the rewards of the arms. However, while the reward realization is immediately known after each pull, value contributions need to be estimated as the worker's ability in a real LCS scenario. Since M_k in LCS corresponds to a limit on the number of times an arm can be pulled, and c_k corresponds to the cost of pulling each arm in MAB, it follows that our problem corresponds to an arm-limited, budget-limited MAB where the realizations of the rewards depend on the workers' ability.

The regret of an algorithm A for a given budget B is defined as:

$$R^A(B) = v^*(B) - v^A(B)$$

where $v^*(B)$ is the optimal solution of the BKP in (3) and $v^A(B)$ is the aggregated value contributions using algorithm A.

3 RELATED WORK

Several heuristic algorithms have been proposed for labeling tasks in CS, however, the performance of these inference algorithms is typically intractable [13, 24, 27]. In [15], an algorithm was proposed for the evaluation of homogeneous labeling tasks, i.e., all the tasks are equally difficult to label. It was proved that the algorithm is order optimal in the number of evaluations required per task required to obtain a desired classification error, when the number of tasks and workers tends to infinity. Thus, the work concluded that using an adaptive algorithm for task assignment has no gain in traditional CS. The model studied in [15] was generalized to heterogeneous labeling tasks in [11]. In this case, the authors showed that adaptive assignment of tasks leads to significant gains both in theory and practice. Unlike our work, the solution in this work is limited to weighted MV and binary labeling of the tasks. In addition, [17] presented tight achievable lower bounds for heterogeneous labeling tasks and proposed an order optimal scheme. For a similar model, [25] exploits the notions of iterative improvement and redundancy for translation tasks outsourced to CS. The work in [12] proposed an online task assignment scheme based on exploration and exploitation for heterogenous tasks. Their system model is budget constrained by assigning a limit on the number of evaluations for each task.

All of the above works consider equal incentives for all the workers and minimize the number of evaluations required per task. However, in a real life scenario a more efficient worker would expect higher incentives for his or her work. Our model allows for different costs per worker, and plans the assignment of the tasks accordingly. Additionally, the model also accounts for a maximum number of tasks that can be performed by a worker.

The Multi-Armed Bandits (MAB) problem is closely related to our crowdsourcing problem. A variety of budget constrained models have been studied in the MAB setup [3, 7, 10]. These works consider a budget-limited exploration in the initial phase followed by a cost-free exploitation phase. However, in a real world setting such as the one considered in LCS, the exploitation phase is not free of cost. This limitation is addressed in the budget-limited MAB problem, where both the exploration and exploitation phase are limited by a single budget. This model also considers different costs for arm selection. Two different policies were proposed in this setting, called ϵ-first policy and KUBE [21, 22]. However, they did not consider a limit on the number of times an arm can be pulled, which is analogous to limiting the number of tasks a worker can do in LCS.

Later, Tran-Thanh et al. extended the ϵ-first policy from [21] to an arm-limited, budget-limited MAB and they showed that the regret of this new policy is $O(B^{2/3})$, where B is the budget[23]. Auer et al. showed that a lower bound on the regret for any algorithm is of the order $\Omega(\log B)$ [4]. It follows that the extended ϵ-first policy is not optimal. Tran-Thanh et al. also extended the KUBE algorithm to arm-limited, budget-limited MAB [23]. However, they do not provide any theoretical performance analysis for this new algorithm, called B-KUBE. In this paper, we show that B-KUBE is indeed order optimal, achieving the lower bound presented in [4].

Other extensions of the MAB setup to CS have been considered in the literature. Abraham et al. studied the workers' selection criteria for different cost of workers [1]. Donmez et al. and Zheng et al. proposed schemes for learning the ability of the labelers for equal and unequal incentives, respectively [9, 26]. Unlike our model, their system is not task limited by the workers and budget limited by the task master. BTASC is a workers' selection scheme proposed for spatial CS, however, it does not have any theoretical guarantees[19]. It is sub-optimal compared to BKUBE as it does not account for different costs paid to the workers. Also,the computation complexity of the scheme is $O(BK^2)$, whereas, the computation complexity for B-KUBE is $O(BK\log(K))$.

Also, there has been a large amount of work on bandits with knapsack [2, 5, 6, 8, 11]. Badanidiyuru et al., Biswas et al., Ding et al., Ho et al. focus on unbounded multidimensional knapsack problem in MAB, whereas, our work studies the bounded knapsack problem (BKP) [5, 6, 8, 11]. In other words, the setup of these works do not consider a limit on the number of tasks that can be performed by each worker. In [5] and [11], workers arrive sequentially, and the workers' selection policy has to be of the form "take-it" or "leave-it". Therefore, unlike LCS, no worker is accessible later for task assignment once left. Agrawal and Devanur assume the constraints of the knapsack problem form a simplex [2]. Therefore, the focus is on a perfectly convex knapsack problem. Unlike our work, this problem setup does not capture the limit on the number of tasks that can be performed by each worker which is an integer programming problem. Additionally, upper confidence bounds proposed in [2, 8] are different than the one used in B-KUBE. Extension of the policy proposed in [6] to BKP setting is of the form of Bounded ϵ-F policy which is suboptimal with respect to BKUBE, and its performance bounds cannot be improved [23].

4 WORKERS' SELECTION

We perform workers' selection using B-KUBE, which is described in Algorithm 1, where n denotes the iteration for worker's selection, B_n is the remaining budget before the n^{th} iteration, m_k is the remaining number of tasks a worker can perform, and $i(n)$ is the worker selected in the n^{th} iteration.

In each iteration, the task master checks the feasibility of worker's selection, i.e., whether there exists a $k \in [K]$ such that $c_k \leq B_n$ and $m_k > 0$. The first K iterations of B-KUBE constitute the initialization phase, where all the workers are selected once. For the remaining iterations, B-KUBE selects a worker j with probability $m^*_{j,n}/\sum_k m^*_{k,n}$, where $m^*_{j,n}$ is the number of selections of worker j proposed by the density-ordered greedy algorithm (DGA) for BKP at the n^{th} iteration.

DGA for BKP is described in Algorithm 2. It gives the number of selections of the workers for the remaining budget B_n. The algorithm computes the upper confidence bound value contribution $\hat{w}_k$, using the estimated value contribution $\hat{v}_k$, as

$$\hat{w}_k = \hat{v}_k + \sqrt{\frac{2\log(n)}{M_k - m_k}},$$

and utilizes the entire B_n to select the workers as many times as possible, taking into account their individual limit m_k at the n^{th} iteration. The workers are selected in decreasing order of their estimated efficiencies $\hat{e}_k = \hat{w}_k/c_k$.

To analyze the performance of B-KUBE, we assume that the budget $\sum_k c_k < B \leq \sum_k c_k M_k$, the value contribution v_k has support in $[0, 1]$, and the cost $c_k \geq 1\ \forall k \in [K]$. All results can easily be generalized using an appropriate scaling factor.

We start by recalling some results from the literature of BKP that are useful in our setting. The BKP formulation in (3) can be relaxed to the linear problem LP-BKP

$$\max_{\{x_k\}} \sum_k x_k v_k \text{ s.t.} \quad \sum_k x_k c_k \leq B, \quad \forall k \in [K] : 0 \leq x_k \leq M_k. \tag{4}$$

The following lemma provides the optimal workers' selection strategy for LP-BKP.

Lemma 4.1. [16]. *If the workers are sorted in decreasing order of their efficiencies $e_k = v_k/c_k$, where $e_1 \geq e_2 \geq \ldots \geq e_K$, then the optimal workers' selection strategy for LP-BKP is*

$$x_k^* = \begin{cases} M_k & \forall k = 1, 2, \ldots, s-1 \\ \frac{B - \sum_{k=1}^{s-1} c_k M_k}{c_s} & k = s \\ 0 & \forall k = s+1, \ldots, K, \end{cases} \tag{5}$$

where the splitting worker s is such that $\sum_{k=1}^{s-1} c_k M_k \leq B$ and $\sum_{k=1}^{s} c_k M_k > B$. The maximum aggregated value contribution is

$$v_{LP-BKP}^* = \sum_{k=1}^{s} x_k^* v_k.$$

Letting v_{BKP}^* be the maximum aggregated value contributions that can be obtained from BKP and v' be the aggregated value contribution corresponding to the selection strategy $\lfloor x^* \rfloor = (x_1^*, x_2^*, \ldots, \lfloor x_s^* \rfloor, 0, 0 \ldots)$, by Lemma 4.1 we have

$$v' \leq v_{BKP}^* \leq v_{LP-BKP}^* \leq v' + v_s. \tag{6}$$

The key idea for obtaining a regret bound for B-KUBE is now to determine the number of times a worker k is selected more than the number of selections of worker k as proposed by $\lfloor x^* \rfloor$. This will provide a bound on the regret of B-KUBE assuming $\lfloor x^* \rfloor$ is the optimal workers' selection strategy. This bound can then be combined with (6) to obtain the regret bound for B-KUBE.

It is worth pointing out the main challenges for the theoretical evaluation of B-KUBE compared to that of KUBE. In the KUBE setup, the computation of the regret bound simply corresponds to determining the expected number of times the most efficient worker is not selected. In the B-KUBE setup, the optimal selection of workers is not limited to a single most efficient worker, and a simplification like the one for KUBE is not possible. We overcome this difficulty by assuming that a feasible solution of BKP is the optimal selection strategy, and bounding the sub-optimal workers' selection based on this assumption. The other challenge is that the selection of the splitting worker s in $\lfloor x^* \rfloor$ is not always optimal. We solve this challenge by giving a bound on the expected number of times a worker k is selected more than the number of selections of worker k as proposed by $\lfloor x^* \rfloor$, as follows

Theorem 4.2. *For a given budget B, let B-KUBE perform N iterations. Assume that $\lfloor x^* \rfloor$ is the optimal selection strategy for the workers. Then, the expected number of times a worker k is selected more than the number of selections proposed by $\lfloor x^* \rfloor$ is*

$$E\left[N_k(N)|N\right] \leq \left(\frac{8}{\min\{Q_{\min}^2, d_s^2\}} + \left(\frac{C_{max}}{C_{min}}\right)^2 \right) \log N + \frac{\pi^2}{3} + 1, \tag{7}$$

where

$$Q_{min} = \min_{k \notin I^* \cup \{s\}} \left| e_k - e_s \right| = \min_{k \notin I^* \cup \{s\}} \left| v_k/c_k - v_s/c_s \right|, \tag{8}$$

I^* is the set of the top $s-1$ workers, arranged in decreasing order of their efficiencies e_k, s is the splitting worker, $d_s = |v_{s-1}/c_{s-1} - v_s/c_s|$, $C_{max} = \max_{k \in [K]} c_k$ and $C_{min} = \min_{k \in [K]} c_k$.

Proof. The proof of this theorem can be found in [20] □

Algorithm 1 Bounded KUBE algorithm

Initialization: $n = 1; B_n = B; m_k = M_k \; \forall k$
while selecting a worker is feasible **do**
  if $n \leq K$ **then**
    Initialization Phase: assign $i(n) = n$
  else
    $\{m_{k,n}^*\}$ =greedyAlgoForBKP($\hat{v}_k, m_k, n, B_n$)
    Choose $i(n)$ with $P(i(n) = j) = \frac{m_{j,n}^*}{\sum_k m_{k,n}^*}$
  end if
  Assign the task to $i(n)$
  Update the value contribution $\hat{v}_{i(n)}$ of $i(n)$
  $B_{n+1} = B_n - c_{i(n)}$
  $m_{i(n)} = m_{i(n)} - 1$
  $n = n + 1$
end while

Algorithm 2 Density Ordered Greedy Algorithm for BKP

Function name: greedyAlgoForBKP
Input: $\hat{v}_k, m_k, n, B_n$
Output:$\{m_{k,n}^*\}$
Initialization: $\hat{w}_k = \hat{v}_k + \sqrt{\frac{2\log(n)}{M_k - m_k}}$, $m_{k,n}^* = 0 \; \forall k$
$\hat{e} = \{e_1, \ldots, e_k\}$ is the list of $(\hat{w}_k, c_k, m_{k,n}^*, m_k)$ sorted in decreasing order with respect to $\hat{w}_k/c_k$
$c = 0$ %the total cost currently used
for $j = 1$ to K **do**
  if $c + \hat{e}(c_j) \leq B_n$ **then**
    assign task to j^{th} worker in $\hat{e}$
    $\hat{e}(m_{j,n}^*) = \min\left(\hat{e}(m_j), \lfloor \frac{B(n) - c}{\hat{e}(c_j)} \rfloor\right)$
    $c = c + \hat{e}(m_{j,n}^*) \cdot \hat{e}(c_j)$
  else
    $\hat{e}(m_{j,n}^*) = 0$
  end if
end for

From Theorem 4.2, it follows that assuming $\lfloor x^* \rfloor$ is the optimal selection strategy , using B-KUBE the selection of sub-optimal workers grows only logarithmically with N and we can conclude that B-KUBE favors the selection of workers as proposed by $\lfloor x^* \rfloor$. Additionally, Q_{min} and d_s measure the minimum separation between the optimal and sub-optimal selections, hence, they are the leading constants of $\log(N)$ in Theorem 4.2. Intuitively, it is more difficult to identify the optimal selection strategy $\lfloor x^* \rfloor$ if the abilities of the workers at the boundary of the optimal and sub-optimal selections are close. Theorem 4.2 recovers the result of the stochastic bandits, which are neither budget limited nor arm limited, with an additional constant factor of one in the leading term $\log(N)$ [4]. The minimum separation between the optimal and sub-optimal selections reduces to the same measure as proposed in [4].

Finally, the following theorem provides the regret bound for B-KUBE

Theorem 4.3. *The regret for B-KUBE is $O(\log(B))$.*

Proof. The proof of this theorem can be found in [20] □

Auer et al. showed that a lower bound on the regret is $\Omega(\log N)$,where N is the total number of iterations [4]. In a budget-limited scenario, the number of iterations N is $\Theta(B)$, since $N \in [B/C_{max}, B/C_{min}]$. It follows that the lower bound on the regret in a budget-limited scenario is $\Omega(\log B)$ and B-KUBE is order optimal for arm-limited, budget-limited MAB.

5 VALUE CONTRIBUTIONS OF WORKERS

At each step n, workers' selection policy discussed in the previous section is dependent on the realization of $i(n)^{th}$ worker's value contribution for the update of its empirical estimate $\hat{v}_{i(n)}$. Therefore, we now focus on the determination of the ability of the workers in terms of value contributions, and propose a strategy for estimating the value contribution in real time.

Let the inference function $f_k(l, \hat{l})$ denote the contribution of the k^{th} worker to the label l when $\hat{l}$ is the label predicted by the k^{th} worker. Then, for all $l \in [L]$, the accumulated contribution to the label l after M evaluations of task i is

$$s_{i,l} = \sum_{n=1}^{M} \sum_{k=1}^{K} f_k(l, \hat{l}_i^{(n)}) y_{k,n}, \tag{9}$$

where $y_{k,n}$ is an indicator function which is unity if the n^{th} evaluation of the task is performed by the k^{th} worker, and $\hat{l}_i^{(n)}$ is the predicted label of task i at n^{th} evaluation. The decision rule is

$$\hat{l}_i = \arg\max_{l \in [L]} s_{i,l}. \tag{10}$$

The inference function $f_k(.,.)$ is assumed to be non-negative, and bounded for all $k \in [K]$. Any generalized inference rule for labeling task is captured by (9) and (10). Special cases include majority voting, weighted majority voting and Maximum A Posteriori (MAP) decision rule.

Two key properties play an important role in the design of the inference function. First, the function should account for the characteristics of an individual worker. For example, if a worker is expected to confuse between the two labels, then the contribution of the inference function to them should be similar when one of these labels is predicted. This knowledge can be acquired from the prior knowledge about the workers' ability, if available. Second, the inference function can be designed by the task master based on the knowledge of the labeling tasks. If two labels are similar to each other, then the contributions to them should be similar, for all the workers, when one of these labels is predicted. Other properties that the task master can consider while designing the inference function are the difficulty level of the tasks and the prior distribution on the labels. Clearly, while all of the above properties can be used to design an appropriate inference function, it is not mandatory to use any these properties. For example, a popular inference rule that does not account for these properties is majority voting (MV), while weighted majority voting takes into account the efficiency of the workers.

The following theorem provides the value contribution of each worker and the relation between the accumulated value contribution and the classification error for each task.

Theorem 5.1. *Given a task i, for the inference rule in (9) and (10), the value contribution v_k for the k^{th} worker is*

$$v_k(l_i^*) = \min_{l \neq l_i^*} \mathbb{E}_{l_i^*} \Big[f_k(l_i^*, Y) - f_k(l, Y) \Big]. \tag{11}$$

Additionally, the classification error $\epsilon_i = \mathbb{P}(\hat{l}_i \neq l_i^*)$ and the accumulated value contribution after M evaluations of a task are related as

$$\sum_{n=1}^{N} \sum_{k=1}^{K} v_k(l_i^*).y_{k,n} \geq \sqrt{MQ^2 \log \frac{L-1}{\epsilon_i}}, \tag{12}$$

where $Q = \max_{k \in [K]} \max_{l^* \in [L]} \max_{\hat{l} \in [L]} f_k(l^*, \hat{l})$.

Proof. The proof of this theorem can be found in [20] □

In LCS, the value contributions of the workers are unknown and need to be estimated online. The workers' responses are modeled by a stochastic model where a worker k can be assigned a task multiple times and the correct label is predicted each time with probability p_k independent of the past responses of the worker about the task. Therefore, using (11), the estimation of the value contribution in LCS is based on the knowledge of true label of task i l_i^* and the estimate of p_k of the worker k. In practise, the true label l_i^* for a task i is unknown. To circumvent this issue in practical crowdsourcing systems, the ground truth l_i^* is estimated by $\hat{l}_i$ after m^{th} evaluation (10). Following the estimate of l_i^*, we estimate p_k for each worker based on its empirical mean. Let m^{th} evaluation of a task i is assigned to a worker k. The worker k is said to have labeled the task correctly if the predicted label at the m^{th} evaluation $\hat{l}_i^{(m)}$ is the same as $\hat{l}_i$, which is an estimate of l_i^* after m evaluations. Since the probability of predicting the correct label is independent of the true label, the empirical estimate of p_k is then updated as the ratio of correctly predicted labels to the total number of evaluations performed by the worker i.e.

$$\hat{p}_k = \frac{\hat{p}_k \cdot \sum_{n=1}^{m-1} y_{k,n} + \mathbb{1}_{\{\hat{l}_i^{(m)} = \hat{l}_i\}} \cdot y_{k,m}}{\sum_{n=1}^{m} y_{k,n}}.$$

Using the estimate of p_k, the value contribution v_k is estimated according to (11), where the expectation is computed using the empirical estimate of p_k. Under the assumption that the value contribution is independent of the true label i.e. for all $l_i^* \in [L]$ $v_k(l_i^*) = v_k$, the current estimate of the value contribution can be used for the workers' selection in the next iteration.

Now, we briefly re-visit the reformulation of LCS problem in (2) to BKP in (3). The reformulation of LCS problem to BKP is dependent on the inference rule. The average classification error ϵ (1) is the average of ϵ_i. Using Theorem 5.1, for a generalized inference rule in (10), the upper bound on the classification error ϵ_i decays exponentially with the increase in aggregated value contributions from the workers for a task i. Thus, minimizing the ϵ_i can be reformulated as maximizing the aggregated value contributions from the workers for task i. Hence, BKP in (3) follows from LCS problem in (2). The key benefit of the reformulation to BKP is that it provides an insight on the optimal aggregation of two different attributes, cost and value contribution, of the worker, and facilitate their comparison on a single scale defined as efficiency in Lemma 4.1. A similar transformation of the problem for labeling tasks, with different constraints, has been considered earlier for special cases such as weighted majority voting and majority voting [1, 11]. However, we formalize the notion of the value contribution for a generalized form of inference rule which recovers the transformation derived for weighted majority voting and majority voting in the literature as a special case.

6 PERFORMANCE EVALUATION

We now compare the performance of B-KUBE in conjunction with our value contribution estimation strategy, with three benchmark MAB algorithms for workers' selection using the same value contribution estimation strategy in LCS setup. The benchmark algorithms are Bounded ϵ-First (Bounded ϵ-F), Trailsourcing, and Budget-Limited ϵ-First (ϵ-F). Bounded ϵ-F and ϵ-F are described in [23], whereas, trailsourcing is a special case of Bounded ϵ-F. Bounded ϵ-F consists of separate exploration and exploitation phases. It allocates an ϵ fraction of the total budget for exploration to estimate the value contributions of the workers. The exploitation phase in Bounded ϵ-F is a single step assignment phase where the labeling tasks are assigned to the workers based on their estimated value contributions. Trailsourcing is a simpler version of Bounded ϵ-F with only one round of exploration phase i.e. each worker is selected exactly once in the exploration phase. Budget-Limited ϵ-First has the same exploration phase as Bounded ϵ-F but in the exploitation phase it assigns all the tasks to a single worker with maximum estimated efficiency.

Like Bounded ϵ-F, the task assignment schemes studied in the literature of traditional CS are based on learning the quality parameters of the workers in the first stage followed by a single step assignment of the tasks to the workers [1, 11, 12, 14, 19]. These schemes are sub-optimal with respect to Bounded ϵ-F as they do not consider the unequal incentives for the workers. Additionally, Tran-Thanh et al. also argue that the theoretical regret bounds of Bounded ϵ-F cannot be improved for any estimation scheme for quality parameters of the workers[23]. Thus,we limit ourselves to the above mentioned three schemes for the performance comparison. We compare BKUBE directly with Bounded ϵ-F, and show that BKUBE outperforms it both experimentally and theoretically.

LCS is a novel system proposed in this work, therefore, an appropriate real data set is not available for labeling tasks in this setup. Thus,the algorithms are compared in an experimental setup. Additionally, the evaluations in a simulated setup are common for CS as the other schemes proposed in the literature are mostly evaluated in a simulated environment[1, 14, 15, 17]. We perform the comparison in a setup where twenty workers express their interest to perform binary labeling tasks i.e. $K = 20$ and $L = 2$. In this setup, the labels are considered to be equally likely and the tasks are assumed to be equally difficult. The experiments are performed for two different set of workers. In set A, every worker predicts the true label with probability $p_k > 1/2$. The set B contains 15 workers from set A and 5 spammers, i.e, $p_k = 0.5$. MV is used as the inference rule for labeling the tasks. Since MV does not account for any prior information about the labels and the workers, it provides a neutral environment to capture the performance of the algorithms for workers' selection in LCS. By Theorem 5.1, the value contribution v_k of a worker k is $v_k = 2p_k - 1$. In this setup, p_k is randomly chosen from the uniform distribution over the interval $[0.5, 1]$. The value contribution v_k can be computed from p_k. Given v_k, c_k is randomly chosen from the uniform distribution over the interval $[v_k, 1 + v_k]$ as a worker with higher value contribution will expect more incentives.

Assignment of the labeling tasks to the workers is a single step process in all the three benchmark algorithms. Therefore, we evaluate the performance of these algorithms for two different set of tasks with number of tasks $T = 50$ and $T = 100$ in each set and the limit M_k on the number of tasks a workers can perform is $0.6T$ for all the workers. Unlike the benchmark algorithms, B-KUBE evaluates one task at a time and moves to a different task whenever the algorithm is confident that the estimated label of the current task is correct. For the evaluations of B-KUBE, we use the criteria proposed in [1] to move on to the next task. For a given budget, the two key performance measures of the algorithms are: classification error and the number of tasks being performed in LCS. The classification error can be reduced by assigning a task to a large number of workers and aggregating the contributions from the workers to predict the final label of the task. However, this will reduce the number of tasks that can be performed in a limited budget. Thus, there is a trade-off between these two performance measures. The evaluations show that B-KUBE outperforms all the three benchmarks for both the performance measures simultaneously, see Figure 1 and 2.

As the budget B increases, the classification error decreases for all the algorithms. This is expected, since a larger number of evaluations of the labeling tasks can be performed if more budget is available. The key observation is that B-KUBE has the smallest classification error whereas the three benchmark algorithms have a higher classification error even after utilizing the available budget to perform less number of tasks in comparison to B-KUBE. Additionally, the classification error of ϵ-F is close to that of Bounded ϵ-F, however, the number of tasks performed by ϵ-F are less than the number of tasks performed by Bounded ϵ-F. This is because the tasks are only assigned to the most efficient worker estimated during the exploration phase. As a consequence, this limits the number of tasks T performed by ϵ-F (Fig. 1 and 2). Another important observation is that the gap between the classification error of the three benchmark algorithms and B-KUBE reduces as the budget increases. This is because the optimal solution of the BKP includes

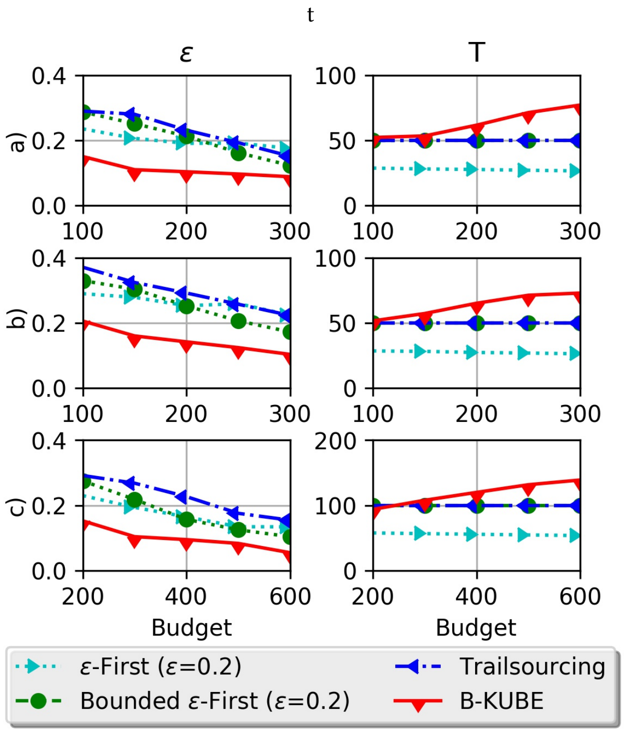

Figure 1: The first and second column of plots are corresponding to the classification error ϵ and number of tasks T performed by the workers respectively. a) T=50 and Set A workers b) T=50 and Set B workers c) T=100 and Set A workers

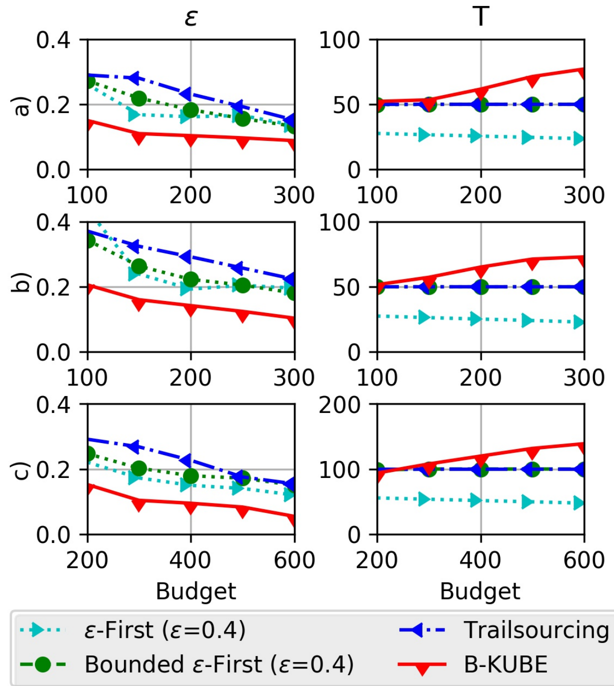

Figure 2: The first and second column of plots are corresponding to the classification error ϵ and number of tasks T performed by the workers respectively. a) T=50 and Set A workers b) T=50 and Set B workers c) T=100 and Set A workers

more and more less efficient workers as the budget increases and the absolute gains from the correct identification of the optimal workers decreases for a large budget. In other words, the losses due to selection of a worker from the sub-optimal set, according to BKP, reduces for large budget.

Figure 1(b) and 2(b) shows the performance of the algorithms in presence of the spammers for the same setting as in Fig 1(a) and 2(a) respectively. An important remark here is that the optimal solution for BKP doesn't include any spammer for the values of B considered in the setup. B-KUBE performs better than the three benchmark algorithms in the presence of spammers as well. However, there is a significant increase in the classification error of the B-KUBE for small budget i.e. $B = 100$. The key reason is the absence of a pure exploration phase in B-KUBE which limits the opportunity to identify the spammers. For large budget $B = 300$, the classification error of B-KUBE does not increase significantly as the algorithm is able to utilize the budget efficiently for the identification of spammers. On contrary, this is not true for the three benchmark algorithms.

In conclusion, B-KUBE has a smaller classification error and performs a larger number of tasks in comparison to the three benchmark algorithms. Note that B-KUBE and Bounded ϵ-F are the DGA based extension of KUBE and ϵ-First policies from a budget-limited MAB setup to an arm-limited, budget-limited MAB setup . Finally, the performance trends of B-KUBE and Bounded ϵ-F in the current setup are similar to the ones of KUBE and ϵ-First policy in a budget-limited MAB setup reported in [22].

7 CONCLUSION

We proposed a notion of Limited-information Crowdsourcing Systems. Unlike traditional CS, LCS monitors the labeling of every single task by a worker in real time, and controls the further assignment of the tasks to the workers based on the estimated value contribution. Due to this form of continuous monitoring, the task master can choose not to assign a task to a worker, and return later to the same worker after exploring other workers, thus, eliminating the requirement of gold-set of tasks. The key challenges in this new setup are determining an efficient workers' selection policy and estimating the value contributions of the workers in real time.

We used B-KUBE to resolve the first challenge and provided its performance analysis, showing that it is an order optimal policy for workers' selection in a budget limited arm limited MAB setup. This work closes the gap in the literature of current MAB setup, showing that B-KUBE is order optimal. To resolve the second challenge, we first introduced the value contributions of the workers for any inference rule and then provided the explicit relation between the accumulated value contribution from the workers and the classification error. We also proposed a strategy to estimate the value contributions of the workers.

We compared the performance of B-KUBE in conjunction with our value contribution estimation strategy, with three benchmark MAB algorithms using the same value contribution estimation strategy in LCS setup. Our experimental evaluations show that B-KUBE outperforms all the three benchmark algorithms for both the performance measures simultaneously. However, it is worth noticing that B-KUBE has a higher computational complexity than the benchmarks evaluated here.

The MAB setup considered in this paper is important as it has extension to various applications like recommendation systems and learning optimal causal intervention. In recommendation systems, the selection of items is analogous to the workers' selection and value of the items need to estimated online from the user's prospective like value contributions in LCS. Likewise, the current MAB setup can be used to learn an optimal causal intervention in Directed Acyclic Graphs. In this application, the intervention selection is analogous to workers' selection and the reward corresponding to the intervention is analogous to workers' value contribution. The budget constraint is applicable to these applications in a similar way as to the current LCS setup. Hence, there exists many applications where the current MAB setup can be used along with an online estimation scheme, depending on the application, to design an efficient multi-agent system. Likewise, the work can be applied to various Multi-agent systems as the budget limited arm limited MAB setup is a popular model for constraining the systems.

Additionally, the work introduces a notion of LCS which triggers another research direction for crowdsourcing systems. The value contributions of the workers can be formulated for more complicated tasks, for example translation and testing, that require variety of skills to complete. If a task requires z skills to be completed then the value contribution of a worker can be modeled as a z dimensional vector where each dimension of the vector corresponds to a particular skill required for completing the task. Designing the workers' selection policy and online strategy to estimate the value contributions of the workers for such tasks is challenging and is left as future work.

REFERENCES

[1] Ittai Abraham, Omar Alonso, Vasilis Kandylas, and Aleksandrs Slivkins. 2013. Adaptive crowdsourcing algorithms for the bandit survey problem. In *Conference on learning theory*. 882–910.

[2] Shipra Agrawal and Nikhil R Devanur. 2014. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*. ACM, 989–1006.

[3] András Antos, Varun Grover, and Csaba Szepesvári. 2008. *Active Learning in Multi-armed Bandits*. Springer Berlin Heidelberg, Berlin, Heidelberg, 287–302. https://doi.org/10.1007/978-3-540-87987-9_25

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[5] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. 2013. Bandits with knapsacks: Dynamic procurement for crowdsourcing. In *The 3rd Workshop on Social Computing and User Generated Content, co-located with ACM EC*.

[6] Arpita Biswas, Shweta Jain, Debmalya Mandal, and Y Narahari. 2015. A truthful budget feasible multi-armed bandit mechanism for crowdsourcing time critical tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1101–1109.

[7] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. 2009. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*. Springer, 23–37.

[8] Wenkui Ding, Tao Qin, Xu-Dong Zhang, and Tie-Yan Liu. 2013. Multi-Armed Bandit with Budget Constraint and Variable Costs.. In *AAAI*.

[9] Pinar Donmez, Jaime G Carbonell, and Jeff Schneider. 2009. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 259–268.

[10] Sudipto Guha and Kamesh Munagala. 2007. Approximation algorithms for budgeted learning problems. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM, 104–113.

[11] Chien-Ju Ho, Shahin Jabbari, and Jennifer W Vaughan. 2013. Adaptive task assignment for crowdsourced classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 534–542.

[12] Chien-Ju Ho and Jennifer Wortman Vaughan. 2012. Online Task Assignment in Crowdsourcing Markets.. In *AAAI*, Vol. 12. 45–51.

[13] Rong Jin and Zoubin Ghahramani. 2003. Learning with multiple labels. In *Advances in neural information processing systems*. 921–928.

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

[15] David R Karger, Sewoong Oh, and Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*. 1953–1961.

[16] H. Kellerer, U. Pferschy, and D. Pisinger. 2004. *Knapsack Problems*. Springer, Berlin, Germany.

[17] Ashish Khetan and Sewoong Oh. 2016. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Advances in Neural Information Processing Systems*. 4844–4852.

[18] Tze Leung Lai and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics* 6, 1 (1985), 4–22.

[19] Chunyan Miao, Han Yu, Zhiqi Shen, and Cyril Leung. 2016. Balancing quality and budget considerations in mobile crowdsourcing. *Decision Support Systems* 90 (2016), 56–64.

[20] Anshuka Rangi and Massimo Franceschetti. 2018. Multi-Armed Bandit Algorithms for Crowdsourcing Systems with Online Estimation of Workers' Ability. *www.arxiv.org* (2018).

[21] Long Tran-Thanh, Archie Chapman, Jose Enrique Munoz De Cote Flores Luna, Alex Rogers, and Nicholas R Jennings. 2010. Epsilon–first policies for budget–limited multi-armed bandits. (2010).

[22] Long Tran-Thanh, Archie C Chapman, Alex Rogers, and Nicholas R Jennings. 2012. Knapsack Based Optimal Policies for Budget-Limited Multi-Armed Bandits.. In *AAAI*.

[23] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. 2014. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence* 214 (2014), 89–111.

[24] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*. 2035–2043.

[25] Omar F Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 1220–1229.

[26] Yaling Zheng, Stephen Scott, and Kun Deng. 2010. Active learning from multiple noisy labelers with varied costs. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 639–648.

[27] Dengyong Zhou, Qiang Liu, John C Platt, Christopher Meek, and Nihar B Shah. 2015. Regularized minimax conditional entropy for crowdsourcing. *arXiv preprint arXiv:1503.07240* (2015).