An Online Learning Approach to Improving the Quality of Crowd-Sourcing

Yang Liu* and Mingyan Liu#
*yangl@seas.harvard.edu, SEAS, Harvard University
#mingyan@umich.edu, EECS, University of Michigan, Ann Arbor

***Abstract*—We consider a crowd-sourcing problem where in the process of labeling massive datasets, multiple labelers with unknown annotation quality must be selected to perform the labeling task for each incoming data sample or task, with the results aggregated using for example simple or weighted majority voting rule. In this paper we approach this labeler selection problem in an online learning framework, whereby the quality of the labeling outcome by a specific set of labelers is estimated so that the learning algorithm over time learns to use the most effective combinations of labelers. This type of online learning in some sense falls under the family of multi-armed bandit (MAB) problems, but with a distinct feature not commonly seen: since the data is unlabeled to begin with and the labelers' quality is unknown, their labeling outcome (or reward in the MAB context) has no way to be verified; thus it can only be estimated against the crowd and known probabilistically. We design an efficient online algorithm LS_OL using a simple majority voting rule that can differentiate high and low quality labelers over time, and is shown to have a regret (w.r.t. always using the optimal set of labelers) of $O(\log^2 T)$ uniformly in time under mild assumptions on the collective quality of the crowd, thus regret free in the average sense. We discuss further performance improvement by using a more sophisticated majority voting rule, and show how to detect and filter out "bad" (dishonest, malicious or very incompetent) labelers to further enhance the quality of crowd-sourcing. Extension to the case when a labeler's quality is task-type dependent is also discussed using techniques from the literature on continuous arms. We establish a lower bound on the order of $O(\log T D_2(T))$, where $D_2(T)$ is an arbitrary function such that $D_2(T) > O(1)$. We further provide a matching upper bound by a minor modification of the algorithm we proposed and studied earlier on. We present numerical results using both simulation and a real dataset on a set of images labeled by Amazon Mechanic Turks (AMT).**

I. Introduction

Machine learning techniques often rely on correctly labeled data for purposes such as building classifiers; this is particularly true for supervised discriminative learning. As shown in [21], [25], the quality of labels can significantly impact the quality of the trained classifier and in turn the system performance. Semi-supervised learning methods, e.g. [5], [15], [30] have been proposed to circumvent the need for labeled data or lower the requirement on the size of labeled data; nonetheless, many state-of-the-art machine learning systems such as those used for pattern recognition continue to rely heavily on supervised learning, which necessitates cleanly labeled data. At the same time, frameworks like participatory sensing rush in enormous quantities of unlabeled data.

An earlier version of this paper appeared at ACM SIGMETRICS'15.

Against this backdrop, crowd-sourcing has emerged as a viable and often favored solution as evidenced by the popularity of the Amazon Mechanical Turk (AMT) system. Prime examples include a number of recent efforts on collecting large scale labeled image datasets, such as ImageNet [8] and LabelMe [24]. The concept of crowd-sourcing has also been studied in contexts other than processing large amounts of unlabeled data, see e.g., user-generated map [10], opinion diffusion [17], and event monitoring [6] in large, decentralized systems.

Its many advantages notwithstanding, the biggest problem with crowd-sourcing is quality control: as shown in several previous studies [12], [25], if labelers (e.g., AMTs) are not selected carefully the resulting labels can be very noisy, due to reasons such as varying degrees of competence, individual difference, and sometimes irresponsible behavior. At the same time, the cost for having large amount of data labeled (payment to the labelers) is non-trivial. This makes it important to look into ways of improving the quality of the crowd-sourcing process and the quality of the results generated by the labelers.

In this chapter we approach the labeler selection problem in an online learning framework, whereby the labeling quality of the labelers is estimated as tasks are assigned and performed, so that an algorithm over time learns to use the more effective combinations of labelers for arriving tasks. This problem in some sense can be cast as multi-armed bandit (MAB) problem, see e.g., [3], [16], [26]. Within such a framework, the objective is to select the best of a set of choices (or "arms") by repeatedly sampling different choices (referred to as *exploration*) and their empirical quality is subsequently used to control how often a choice is used (referred to as *exploitation*). However, there are two distinct features that set our problem apart from the existing literature in bandit problems. Firstly, since the data is unlabeled to begin with and the labelers' quality is also unknown, a particular choice of labelers leads to unknown quality of their labeling outcome (mapped to the "reward" of selecting a choice in the MAB context). Whereas this reward is assumed to be known instantaneously following a selection in the MAB problem, in our model this remains unknown and at best can only be estimated with a certain error probability. This poses significant technical challenge compared to a standard MAB problem. Secondly, to avoid having to deal with a combinatorial number of arms, it is desirable to learn and estimate each individual labeler's quality separately (as opposed to estimating the quality of different combinations of labelers). The optimal selection of labelers

then depends on individual qualities as well as how the labeling outcome is computed using individual labels. In this study we will consider both a simple as well as a weighted majority voting rule and derive the respective optimal selection of labelers given their estimated quality.

Due to its online nature, our algorithm can be used in real time, processing tasks as they arrive. Our algorithm thus has the advantage of performing quality assessment and adapting to better labeler selections as tasks arrive. This is a desirable feature because generating and processing large datasets can incur significant cost and delay, so the ability to improve labeler selection on the fly (rather than waiting till the end) can result in substantial cost savings and improvement in processing quality. Below we review the literature most relevant to the study presented in this paper in addition to the MAB literature cited above.

Within the context of learning and differentiating labelers' expertise in crowd-sourcing systems, a number of studies have looked into offline algorithms. For instance, in [9], methods are proposed to eliminate irrelevant users from a set of user-generated dataset; in this case the elimination is done as post-processing to clean up the data since the data has already been labeled by the labelers (tasks have been performed). Another example is the family of matrix factorization or matrix completion based methods, see e.g., [29], where labeler selection is implicitly done through the numerical process of finding the best recommendation for a participant. Again this is done after the labeling has already been done for all data by all (or almost all) labelers. This type of approaches is more appropriate when used in a recommendation system where data and user-generated labels already exist in large quantities.

Recent studies [13], [14] have examined the fundamental trade-off between labeling accuracy and redundancy in task assignment in crowd-sourcing systems. In particular, it is shown in [14] that a labeling accuracy of $1-\varepsilon$ for each task can be achieved with a per-task assignment redundancy no more than $O(K/q \cdot \log(K/\varepsilon))$; thus more redundancy can be traded for more accurate outcome. In [14] the task assignment is done in a one-shot fashion (thus non-adaptive) rather than sequentially with each task arrival as considered in our paper, thus the result is more applicable to offline settings similar to those cited in the previous paragraph. In [13] an iterative algorithm is proposed for deciding tasks assignment and it is shown to outperform majority voting. Again the approach here is one-shot where all questions are asked simultaneously and the allocation rule is non-adaptive.

Within online solutions, the concept of active learning has been quite intensively studied, where the labelers are guided to make the labeling process more efficient. Examples include [12], which uses a Bayesian framework to actively assign unlabeled data based on past observations on labeling outcomes, and [19], which uses a probabilistic model to estimate the labelers' expertise. However, most studies on active learning require either an oracle to verify the correctness of the finished tasks which in practice does not exist, or ground-truth feedback from indirect but relevant experiments (see e.g., [12]). Similarly, existing work on using online learning for task assignment also typically assumes the availability of ground-truth (as in MAB problems). For instance, in [11] online learning is applied to sequential task assignment but ground-truth of the task performance is used to estimate the performer's quality. In [20], a Bayes update aided online solution was proposed to minimize the regret in a problem of disseminating news to a crowd of users. The performance of the developed algorithm was shown to be better than Thompson Sampling based solutions. However, again, for the setting considered in the above paper, ground-truth signals indicating whether the user likes or dis-likes pushed news are revealed immediately after each dissemination. In this sense, our results cannot be compared directly to those cited above.

Our work differs from the above as we do not require oracle or the availability of immediate ground-truth; we instead impose a mild assumption on the collective quality of the crowd (without which crowdsourcing would be useless and would not have existed), so an estimated or imperfect ground-truth can be inferred. Secondly, our framework allows us to obtain performance bounds on the proposed algorithm in the form of regret with respect to the optimal strategy that always uses the best set of labelers; this type of performance guarantee is lacking in most of the work cited above. Last but not least, our algorithm is very broadly applicable to a generic crowd-sourcing task assignment rather than being designed for specific type of tasks or data.

Our main contributions are summarized as follows. (1) We design an online learning algorithm to estimate the quality of labelers in a crowd-sourcing setting without ground-truth information but with mild assumptions on the quality of the crowd as a whole, and show that it is able to learn the optimal set of labelers under both simple and weighted majority voting rules and attains no-regret performance guarantees (w.r.t. always using the optimal set of labelers). (2) We similarly provide regret bounds on the cost of this learning algorithm w.r.t. always using the optimal set of labelers. (3) We show how our model and results can be extended to the case where the quality of a labeler may be task-type dependent, as well as a simple procedure to quickly detect and filter out "bad" (dishonest, malicious or incompetent) labelers to further enhance the quality of crowd-sourcing. (3) We establish a lower bound on the learning regret for our online labeler selection problem and refine our algorithm to match this lower bound. (4) Our validation includes both simulation and the use of a real-world AMT dataset.

The remainder of this paper is organized as follows. We formulate our problem in Section II. In Sections III and IV we introduce our learning algorithm along with regret analysis under a simple majority and weighted majority voting rule, respectively. Lower bound results on our learning algorithm is presented in Section V, as well as we provide a matching and refined upper bound. Numerical results are presented in Section VI. Section VIII concludes the paper.

II. Problem formulation and preliminaries

A. The crowd-sourcing model

We begin by introducing the following major components of the crowd-sourcing system.

User. There is a single user with a sequence of tasks (unlabeled data) to be performed/labeled. Our proposed on-line learning algorithm is to be employed by the user in making labeler selections. Throughout our discussion the terms *task* and *unlabeled data* will be used interchangeably.

Labeler. There are a total of M labelers, each may be selected to perform a labeling task for a piece of unlabeled data. The set of labelers is denoted by $\mathscr{M} = \{1,2,...,M\}$[1]. A labeler i produces the true label with probability p_i independent of the task, and independent of each other[2]; a more sophisticated task-dependent version is discussed in Section III. This will also be referred to as the quality or accuracy of this labeler. We will assume no two labelers are exactly the same, i.e., $p_i \neq p_j, \forall i \neq j$ and $0 < p_i < 1, \forall i$. These quantities are unknown to the user *a priori*. We will also assume that the accuracy of the collection of labelers satisfies $\sum_{i=1}^{M} \frac{p_i}{M} > \frac{1}{2}$. The justification and implication of this assumption are discussed in more detail in Section II-C.

Our learning system works in discrete time steps $t = 1,2,...,T$. At time t, a task $k \in \mathscr{K}$ arrives to be labeled, where $\mathscr{K}$ could be either a finite or infinite set. For simplicity of presentation, we will assume that a single task arrives at each time, and that the labeling outcome is binary: 1 or 0; however, both assumptions can be fairly easily relaxed[3]. For task k, the user selects a subset $S_t \subseteq \mathscr{M}$ to label it. The label generated by labeler $i \in S_t$ for data k at time t is denoted by $L_i(t)$.

The set of labels $\{L_i(t)\}_{i \in S_t}$ generated by the selected labelers then need to be combined to produce a single label for the data; this is often referred to as the information aggregation phase. Since we have no prior knowledge on the labelers' accuracy, we will apply the simple majority voting rule over the set of labels; later we will also examine a more sophisticated weighted majority voting rule. Mathematically, the majority voting rule at time t leads to the following output: $L^*(t) = \text{argmax}_{l \in \{0,1\}} \sum_{i \in S_t} 1\{L_i(t) = l\}$, with ties (i.e., $\sum_{i \in S_t} 1\{L_i(t) = 0\} = \sum_{i \in S_t} 1\{L_i(t) = 1\}$) broken randomly.

Denote by $\pi(S_t)$ the probability of correct labeling outcome following the simple majority rule above, and we have:

$$\pi(S_t) = \underbrace{\sum_{S: S \subseteq S_t, |S| \geq \lceil \frac{|S_t|+1}{2} \rceil} \prod_{i \in S} p_i \cdot \prod_{j \in S_t \setminus S} (1-p_j)}_{\text{Majority wins}} + \underbrace{\frac{\sum_{S: S \subseteq S_t, |S| = \frac{|S_t|}{2}} \prod_{i \in S} p_i \cdot \prod_{j \in S_t \setminus S} (1-p_j)}{2}}_{\text{Ties broken equally likely}} . \quad (1)$$

Denote by c_i a normalized cost/payment per sample for labeler i and consider the following linear cost function

$$\mathscr{C}(S) = \sum_{i \in S} c_i, \ S \subseteq \mathscr{M} \ . \quad (2)$$

Denote $S^* = \text{argmax}_{S \subseteq \mathscr{M}} \pi(S)$, thus S^* is the optimal selection of labelers given each individual's accuracy. We also refer to $\pi(S)$ as the utility for selecting the set of labelers S and denote it equivalently as $U(S)$. $\mathscr{C}(S^*)$ will be referred to as the *necessary cost* per task.

The design of a good online algorithm may require tradeoff between the labeling accuracy and labeling cost. They can be combined into the same objective by properly weighing the two, e.g., in a linear combination shown below:

$$U_\eta(S) := \pi(S) - \eta \cdot \mathscr{C}(S), \quad (3)$$

where $\eta \geq 0$ is a constant trading-off the labeling accuracy and the total budget. Define the optimal target set as $S^*_\eta = \text{argmax}_{S \subseteq \mathscr{M}} U_\eta(S)$. When $\eta = 0$ the above reduce to simply the labeling accuracy $U(S)$ and set S^*, respectively, and we will thus use these and $U_{\eta=0}(\cdot), S^*_{\eta=0}$ interchangeably. While our analysis primarily focuses on learning S^* by treating accuracy and cost separately, it extends to $S^*_{\eta>0}$ in a fairly straightforward manner and thus our main result is for the general case of $U_\eta(\cdot)$ and S^*_η.

B. Offline optimal selection of labelers

Before addressing the learning problem, we will first take a look at how to efficiently derive the optimal selection S^* given accuracy probabilities $\{p_i\}_{i \in \mathscr{M}}$. This will be a crucial step repeatedly invoked by the learning procedure we develop next, to determine the set of labelers to use given a set of *estimated* accuracy probabilities.

The optimal selection is a function of the values $\{p_i\}_{i \in \mathscr{M}}$, the aggregation rule used to compute the final label, and the unit cost c_i's. While there is a combinatorial number of possible worker selections, the next two results combined lead to a simple and linear-complexity procedure in finding the S^*_η, when $\eta = 0$, the accuracy and cost are evaluated separately; or $\eta > 0$ and $c_i \equiv c$ that workers have the same cost. Note we do not need $c_i \equiv c$ & $\eta = 0$; and assuming $c_i \equiv c$ is not entirely a simplification – on popular crowdsourcing markets, e.g. AMT, workers are typically paid an equal amount for the same task; and such an assumption has been widely adopted in crowdsourcing studies [7], [28]. It is also worth noting that in practice the payment is set by the user who does not differentiate price when assigning the tasks because the quality or experience of the labelers are unknown a priori. .

Theorem 1: Under the simple majority vote rule, the optimal number of labelers $s^*_\eta = |S^*_\eta|$ must be an odd number.

Theorem 2: The optimal set S^*_η is monotonic, i.e., if we have $i \in S^*_\eta$ and $j \notin S^*_\eta$ then we must have $p_i > p_j$.

Proofs of the above two theorems can be found in [18]. Based on above results given a set of accuracy probabilities, the optimal selection under the majority vote rule consists of the top s^*_η (an odd number) labelers with the highest quality; we only need to compute s^*_η, which has a linear complexity of $O(M/2)$. A set that consists of the highest m labelers will be referred to as a *m-monotonic set*, and denoted as $S^m \subseteq \mathscr{M}$.

When $\eta > 0$ and c_is differ, the optimal set can be very different and is subject to further study. [4]

[1]We could also model M types of workers, each arriving in sequence.

[2]Assumption of independence is made to simplify the analysis. In practice, when labelers are correlated, tools from correlated MAB can be used.

[3]Indeed in our experiment shown later in Section VI, our algorithm is applied to a non-binary multi-label case.

[4]This does not affect the learning structure we study later. However, in this case the offline optimal solution will lose the clear structural property, which then requires the learning to be more fine grained in order to differentiate all possible combinations of labelers; this in turn will lead to higher regret.

C. The lack of ground-truth

As mentioned, a key difference between our model and many other studies on crowdsourcing as well as the basic framework of MAB problems is that we lack ground-truth in our system; we elaborate on this below. In a standard MAB setting, when a player (the user in our scenario) selects a set of arms (labelers) to activate, she immediately finds out the rewards associated with those selected arms. This information allows the player to collect statistics on each arm (e.g., sample mean rewards) which is then used in her future selection decisions. In our scenario, the user sees the labels generated by each selected labeler, but does not know which ones are true. In this sense the user does not find out about her reward immediately after a decision; she can only do so probabilistically over a period of time through additional estimation devices. This constitutes the main conceptual and technical difference between our problem and standard MAB.

Given the lack of ground-truth, the crowdsourcing system is only useful if the average labeler is more or less trustworthy. For instance, if a majority of the labelers produce the wrong label most of the time, unbeknownst to the user, then the system is effectively useless, i.e., the user has no way to tell whether she could trust the outcome so she might as well abandon the system. It is therefore reasonable to have some trustworthiness assumption in place. Accordingly, we shall assume that $\bar{p} := \sum_{i=1}^{M} p_i/M > 1/2$, i.e., the average labeling quality is higher than 0.5; this is a common assumption in the crowd-sourcing literature (see e.g., [9]). Note that this is a fairly mild assumption: not all labelers need to have accuracy $p_i > 0.5$ or near 0.5; some labeler may have arbitrarily low quality (~ 0) as long as it is in the minority. When $\bar{p} \leq 1/2$, which is often referred to as the case when majority people are wrong [22], it is possible to apply certain machine learning approach on the set of collected labels to determine whether evidence exists indicating the crowd is on the whole misleading. More advanced and formal reporting mechanisms have been developed to elicit the true answer, see e.g., the well known Bayesian Truth Serum (BTS) algorithm (again see [22]). However, such mechanisms usually require reporting more information besides the label output. In this study we shall limit ourselves to the simpler case $\bar{p} > 0.5$ whereby only label output needs to be reported; similar problems for the case $\bar{p} < 0.5$ warrants a separate study. Denote by X_i a binomial random variable with parameter p_i to model labeler i's outcome on a given task: $X_i = 1$ if her label is correct and 0 otherwise. Using Chernoff Hoeffding's inequality we have

$$P(\frac{\sum_{i=1}^{M} X_i}{M} > 1/2) = 1 - P(\frac{\sum_{i=1}^{M} X_i}{M} \leq 1/2) \geq 1 - e^{-2M\cdot(\bar{p}-1/2)^2}.$$

Define $a_{\min} := P(\frac{\sum_{i=1}^{M} X_i}{M} > 1/2)$; note this is the probability that a simple majority vote over the M labelers is correct. Therefore, if $\bar{p} > 1/2$ and further $M > \frac{\log 2}{2(\bar{p}-1/2)^2}$, then $1 - e^{-2M\cdot(\bar{p}-1/2)^2} > 1/2$, meaning a simple majority vote would be correct most of the time. Throughout the paper we will assume both these conditions are true. We will also have the following fact: $P(\frac{\sum_{i\in S^*} X_i}{|S^*|} > 1/2) \geq P(\frac{\sum_{i=1}^{M} X_i}{M} > 1/2)$; the inequality is due to the definition of the optimal set S^*.

III. Learning the Optimal Labeler Selection

In this section we present an online learning algorithm LS_OL that over time learns each labeler's accuracy, which it then uses to compute an estimated optimal set of labelers using the properties given in the previous section.

A. An online learning algorithm LS_OL

The algorithm consists of two types of time steps, exploration and exploitation, as is common to online learning. However, the exploration step design is complicated by the additional estimation due to the lack of ground truth revelation. Specifically, a set of tasks will be designated as "testers" and may be repeatedly assigned to the same labeler in order to obtain sufficient results used for estimating her label quality. This can be done in one of two ways depending on the nature of the tasks. For tasks like survey questions (with binary answers), a labeler may indeed be prompted to answer the same question (or equivalent variants with alternative wording) multiple times, usually not in succession, during the survey process. This is a common technique used by survey designers for quality control by testing whether a participant answers questions randomly or consistently, whether a participant is losing attention over time, and so on, see e.g., [23]. For tasks like labeling images, a labeler may be given identical images repeatedly or each time with added small iid noise.

With the above in mind, the algorithm conceptually proceeds as follows. A condition is checked to determine whether the algorithm should explore or exploit in a given time step. If it is to exploit, then the algorithm selects the best set of labelers based on current quality estimates to label the arriving task. If it is to explore, then the algorithm will either assign an old task (an existing tester) or the new arriving task (which then becomes a tester) to the set of labelers $\mathscr{M}$ depending on whether all existing testers have been labeled enough number of times. Because of the need to repeatedly assign an old task, some new tasks will not be immediately assigned (those arriving during an exploration step while an old task remains under-sampled). These tasks will simply be given a random label (with error probability $1/2$) but their numbers are limited by the frequency of an exploration step ($\sim \log^2 T$).

Before proceeding to a more precise description of the algorithm, a few additional notions are in order. Denote the n-th label outcome (via majority vote over M labelers in exploration) for task k by $y_k(n)$. Denote by $y_k^*(N)$ the label obtained using majority rule over the N label outcomes $y_k(1), y_k(2), \cdots, y_k(N)$, and $1\{\cdot\}$ as the indicator function:

$$y_k^*(N) = \begin{cases} 1, & \sum_{n=1}^{N} 1\{y_k(n) = 1\}/N > 0.5 \\ 0, & \text{otherwise} \end{cases}, \qquad (4)$$

with ties broken randomly. It is this majority label after N tests on a tester task k that will be used to analyze different labeler's performance. As we show later in algorithm design, a tester task is always assigned to all labelers for labeling. Therefore these repeated outcomes $y_k(1), y_k(2), \cdots, y_k(N)$ are of the same statistical quality. We will additionally impose the assumption that these outcomes are also independent.

Denote by $E(t)$ the set of tasks assigned to the M labelers during explorations up to time t. For each task $k \in E(t)$ denote by $\hat{N}_k(t)$ the number of times k has been assigned. Consider the following random variable defined at each time t:

$$\mathscr{O}(t) = 1\{|E(t)| \leq D_1(t) \text{ or } \exists k \in E(t) \text{ s.t. } \hat{N}_k(t) \leq D_2(t)\} \ ,$$
$$\text{where } \ D_1(t) = [(\frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)} - \alpha) \cdot \varepsilon_\eta]^{-2} \cdot \log t \ ,$$
$$D_2(t) = (a_{\min} - 0.5)^{-2} \cdot \log t \ ,$$

and $n(S^m)$ is the number of all possible majority subsets (for example when $|S^m| = 5$, $n(S^m)$ is the number of all possible subset of size being at least 3 of S^m, ε_η a bounded constant that depends on η[5], and α a positive constant such that $\alpha < \frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)}$. Note that $\mathscr{O}(t)$ captures the event whether an insufficient number of tasks have been assigned under exploration or whether any task has been assigned insufficient number of times in exploration. We provide some intuitions on the above parameter setting. In inspecting the terms that make up $D_1(t)$, we note that ε is used to bound the gap between the optimal and sub-optimal utility values, while $\varepsilon / \max_{m:m \text{ odd}} m \cdot n(S^m)$ gives an upper bound on the error in estimating each p_i in each majority voting term. The α fraction of loss in the gap is due to the noisy inference of the ground-truth label. The selection of $D_2(t)$ is to make sure the aggregated labels returned by the re-exploration/test phases is sufficiently accurate. This quantity depends heavily on the average accuracy of the entire crowd quantified by $a_{\min} - 1/2$.

Online Labeler Selection: LS_OL

1: Initialization at $t = 0$: Initialize the estimated accuracy $\{\tilde{p}_i\}_{i \in \mathscr{M}}$ to some value in $[0,1]$; denote the initialization task as k, set $E(t) = \{k\}$ and $\hat{N}_k(t) = 1$.
2: At time t a new task arrives: If $\mathscr{O}(t) = 1$ ($\eta = 0$), the algorithm explores.
  2.1: If there is no task $k \in E(t)$ such that $\hat{N}_k(t) \leq D_2(t)$, then assign the new task to $\mathscr{M}$ and update $E(t)$ to include it and denote it by k; if there is such a task, randomly select one of them, denoted by k, to $\mathscr{M}$. $\hat{N}_k(t) := \hat{N}_k(t) + 1$; obtain the label $y_k(\hat{N}_k(t))$;
  2.2: Update $y_k^*(\hat{N}_k(t))$ (using the alternate indicator function $I(\cdot)$): $y_k^*(\hat{N}_k(t)) = 1\{\frac{\sum_{\hat{t}=1}^{\hat{N}_k(t)} y_k(\hat{t})}{\hat{N}_k(t)} > 0.5\}$.
  2.3: Update labelers' accuracy estimate $\forall i \in \mathscr{M}$:
$$\tilde{p}_i = \frac{\sum_{k \in E(t), k \text{ arrives at time } \hat{t}} 1\{L_i(\hat{t}) = y_k^*(\hat{N}_k(t))\}}{|E(t)|} \ .$$
3: Else if $\mathscr{O}(t) = 0$, the algorithm exploits and computes: $S_t = \text{argmax}_m \tilde{U}(S^m)$ which is solved using the linear search property, but with the current estimates $\{\tilde{p}_i\}$ rather than the true quantities $\{p_i\}$, resulting in estimated utility $\tilde{U}()$ and $\tilde{\pi}()$. Assign the new task to those in S_t.
4: Set $t = t + 1$ and go to Step 2.

Fig. 1: Description of LS_OL

[5]We will similarly use ε to denote $\varepsilon_{\eta=0}$

Our online algorithm for labeler selection is formally shown in Fig. 1, for the case of $\eta = 0$. This can be easily adapted to the general case by changing Step 3 to $S_t = \text{argmax}_m \tilde{U}_\eta(S^m)$.

The above algorithm can either go on indefinitely or terminate at some time T. As we show below the performance bound on this algorithm holds uniformly in time so it does not matter when it terminates. Note the search complexity at each step t is simply $O(N)$, following the optimality of linear search established in Theorem 2. The computational complexity for computing $\pi(S)$ for each S is combinatorial in $|S|$.

B. Main results

The standard metric for evaluating an online algorithm in the MAB literature is *regret*, the difference between the performance of an algorithm and that of a reference algorithm which often assumes foresight or hindsight. The most commonly used is the *weak regret* measure with respect to the best single-action policy assuming a priori knowledge of the underlying statistics. In our problem context, this means to compare our algorithm to the one that always uses the optimal selection S^*. It follows that this weak regret, up to time T, is given by

$$R_\eta(T) = T \cdot U(S_\eta^*) - E[\sum_{t=1}^{T} U(S_t)] \ ,$$
$$R_{\mathscr{C}}(T) = E[\sum_{t=1}^{T} \mathscr{C}(S_t)] - T \cdot \mathscr{C}(S^*) \ ,$$

where S_t is the selection made at time t by our algorithm; if t happens to be an exploration then $S_t = \mathscr{M}$. $R_\eta(T)$ captures the regret for the learning algorithm while $R_{\mathscr{C}}(T)$ is the one for cost. Define: $\Delta_{\max} = \max_{S^m \neq S^*} U(S^*) - U(S^m)$, $\delta_{\max} = \max_{i \neq j} |p_i - p_j|$, $\Delta_{\min} = \min_{S^m \neq S^*} U(S^*) - U(S^m)$, $\delta_{\min} = \min_{i \neq j} |p_i - p_j|$. ε is a constant such that $\varepsilon < \min\{\frac{\Delta_{\min}}{2}, \frac{\delta_{\min}}{2}\}$. Similarly for $\eta > 0$ define $\Delta_{\max,\eta} := \max_{S^m \neq S_\eta^*} U_\eta(S_\eta^*) - U(S^m), \Delta_{\min,\eta} := \min_{S^m \neq S_\eta^*} U(S_\eta^*) - U(S^m)$, and define $\varepsilon < \min\{\frac{\Delta_{\min,\eta}}{2}, \frac{\delta_{\min}}{2}\}$ For analysis we assume $U(S^i) \neq U(S^j)$ if $i \neq j$. Define the sequence $\{\beta_n\}$: $\beta_n = \sum_{t=1}^{\infty} \frac{1}{t^n}$. Our main theorem is stated as follows, when $\eta = 0$ or $\eta > 0$ & $c_i \equiv c$:

Theorem 3: The regrets can be bounded uniformly in time:

$$R_\eta(T) \leq \frac{U_\eta(S_\eta^*) \cdot \log^2(T)}{(\frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon_\eta^2 \cdot (a_{\min} - 0.5)^2}$$
$$+ \Delta_{\max,\eta} (2 \sum_{m=1, m \text{ odd}}^{M} m \cdot n(S^m) + M) \cdot (2\beta_2 + \frac{\beta_{2-z}}{\alpha \cdot \varepsilon_\eta}) \ , \quad (5)$$

$$R_{\mathscr{C}}(T) \leq \frac{\sum_{i \in \mathscr{M}} c_i \cdot \log^2(T)}{(\frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2 \cdot (a_{\min} - 0.5)^2}$$
$$+ \frac{\sum_{i \notin S^*} c_i \cdot \log T}{(\frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2}$$
$$+ (M - |S^*|) \cdot (2 \sum_{m=1, m \text{ odd}}^{M} m \cdot n(S^m) + M) \cdot (2\beta_2 + \frac{\beta_{2-z}}{\alpha \cdot \varepsilon}) \ , \quad (6)$$

where $0 < z < 1$ is a positive constant.

Again, when $\eta = 0$ the above are bounds on accuracy and cost separately. First note that the regret is nearly logarithmic in T and therefore it has zero average regret as $T \to \infty$; such an algorithm is often referred to as a zero-regret algorithm. Secondly the regret bound is inversely related to the minimum accuracy of the crowd (through $a_{\min}$). This is to be expected: with higher accuracy (a larger $a_{\min}$) of the crowd, crowdsourcing generates ground-truth outputs with higher probability, and hence the learning process could be accelerated. Finally, the bound also depends on $\max_m m \cdot n(S^m)$ which is roughly on the order of $O(\frac{2^m\sqrt{m}}{\sqrt{2\pi}})$.

C. Regret analysis of LS_OL

We now outline key steps in the proof of the above theorem. This involves a sequence of lemmas; the proofs of most can be found in the appendix. There are a few that we omit for brevity; in those cases sketches are provided. As can be seen from the proof, the two cases $\eta = 0$ and $\eta > 0$ are conceptually the same. Thus for simplicity of presentation our analysis will first focus on the case of $\eta = 0$, and then illustrate how the results can generalize to a combined utility function with $\eta > 0$ and $c_i \equiv c$.

Step 1: We begin by noting that the regret consists of that arising from the exploration phase and from the exploitation phase, denoted by $R_e(T)$ and $R_x(T)$, respectively:

$$R(T) = E[R_e(T)] + E[R_x(T)] \ .$$

The following result bounds the first element of the regret.

Lemma 1: The regret up to time T from the exploration phase can be bounded as follows:

$$E[R_e(T)] \leq U(S^*) \cdot (D_1(T) \cdot D_2(T)) \ . \tag{7}$$

We see the regret depends on the exploration parameters as product. This is because for tasks arriving in exploration steps, we assign it at least $D_2(T)$ times to the labelers; each time when re-assignment occurs, a new arriving task is given a random label while under an optimal scheme each missed new task means a utility of $U(S^*)$.

Step 2: We now bound the regret arising from the exploitation phase as a function of the number of times the algorithm uses a sub-optimal selection when the ordering of the labelers is correct, and the number of times the estimates of the labelers' accuracy result in a wrong ordering. The proof of the lemma below is omitted as it is fairly straightforward.

Lemma 2: For the regret from exploitation we have:

$$E[R_x(T)] \leq \Delta_{\max} \left(E[\sum_{t=1}^{T} (\mathscr{E}_1(t) + \mathscr{E}_2(t))] \right) \ . \tag{8}$$

Here $\mathscr{E}_1(t) = I_{S_t \neq S^*}$, conditioned on correct ordering of labelers, records whether the a sub-optimal section (other than S^*) was used at time t based on the current estimates $\{\tilde{p}_i\}$. $\mathscr{E}_2(t)$ records whether at time t the set $\mathscr{M}$ is sorted in the wrong order because of erroneous estimates $\{\tilde{p}_i\}$.

Step 3: We proceed to bound the two terms in (8) separately. In this part of the analysis we only consider those times t when the algorithm exploits.

Lemma 3: At time t we have:

$$E[\mathscr{E}_1(t)] \leq \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} m \cdot n(S^m) \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \tag{9}$$

The idea behind the above lemma is to use a union bound over all possible events where the wrong set is chosen when the ordering of the labelers is correct according to their true accuracy.

Lemma 4: At time t we have: $E[\mathscr{E}_2(t)] \leq M(\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}})$

Step 4: Summing up all results and rearranging terms lead to the theorem. Specifically,

$$\begin{aligned} E[R_x(T)] &\leq \Delta_{\max} \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} 2 \sum_{t=1}^{T} m \cdot n(S^m) \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \\ &\quad + \Delta_{\max} \cdot M \cdot \sum_{t=1}^{T} (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \\ &\leq \Delta_{\max} (2 \cdot \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} m \cdot n(S^m) + M) \cdot (2\beta_2 + \frac{1}{\alpha \cdot \varepsilon} \beta_{2-z}) \ . \end{aligned}$$

Since $\beta_{2-z} < \infty$ for $z < 1$, we have bounded the exploitation regret by a constant. This result also implies that if the number of assignments is not a concern (i.e., we can afford to assign tasks to all labelers each time), we will be able to bound the regret on labeling accuracy up to a constant.

Step 5: Summing over all terms in $E[R_e(T)]$ and $E[R_x(T)]$ we obtain the main theorem. We now argue that bounding for the general case η ($R_\eta(T)$, but with $c_i \equiv c$ when $\eta > 0$) follows the same line of proof, due to the fact that the estimation error for the combined utility function for each set S is exactly the same as the estimation error for the labeling accuracy: $\tilde{U}_\eta(S) - U_\eta(S) = \tilde{\pi}(S) - \pi(S)$, which is independent of η. This allows us to first establish the bound in the case of $\eta = 0$ ($R(T)$), and the extension to $R_\eta(T)$ follows straight-forwardly; the proof is thus omitted.

D. Cost analysis of LS_OL

We now analyze the cost regret. Following similar analysis we first note that it can be calculated separately for the exploration and exploitation steps. For *exploration* steps we know the cost regret is bounded by

$$\sum_{i \notin S^*} c_i \cdot D_1(T) + \sum_{i \in \mathscr{M}} c_i \cdot D_1(T) \cdot (D_2(T) - 1)$$

where the second term is due to the fact for all costs associated with task re-assignments are treated as additional costs.

For *exploitation* the additional cost is upper-bounded by

$$(M - |S^*|) \cdot E[\sum_{t=1}^{T} (\mathscr{E}_1(t) + \mathscr{E}_2(t))] \ .$$

Based on previous results we know the cost regret $R_{\mathscr{C}}(T)$ will look similar to $R(T)$ with both terms bounded by either a log term or a constant. Plug in $D_1(T), D_2(T), E[\sum_{t=1}^{T} (\mathscr{E}_2(t)]$, $E[\sum_{t=1}^{T} \mathscr{E}_2(t)]$ we establish the regret for $R_{\mathscr{C}}(T)$ as claimed in our main result.

E. Discussion

We end this section with a discussion on how to relax a number of assumptions adopted in our analytical framework.

IID re-assignments: The first concerns the re-assignment of the same task (or iid copies of the same task) and the assumption that the labeling outcome each time is independent. In the case where iid copies are available, this assumption is justified. In the case when the exact same task must be re-assigned, enforcing a delay between successive re-assignments can make this assumption more realistic. Suppose the algorithm imposes a random delay τ_k, a positive random variable uniformly upper-bounded by $\tau_k \leq \tau_{\max}, \forall k$. Then following similar analysis we can show the upper bound for regret is at most $\tau_{\max}$ times larger, i.e., it can be bounded by $\tau_{\max} \cdot R(T)$, where $R(T)$ is as defined in Eqn. (5).

Prior knowledge of several constants: The second assumption concern the selection of constant ε by the algorithm and the analysis which requires knowledge on $\Delta_{\min}$ and $\delta_{\min}$. This assumption however can be removed by using a decreasing sequence ε_t. This is a standard technique that has been commonly used in the online learning literature, see e.g., [26]. Specifically, let $\varepsilon_t = \frac{1}{\log^{\eta}(t)}$, for some $\eta > 0$. Replace $\log(t)$ with $\log^{1+2\eta}(t)$ in $D_1(t)$ and $D_2(t)$ it can be shown that there exists T_0 s.t. $\varepsilon_{T_0} < \varepsilon$. Thus the regret associated with using an imperfect ε_t is bounded by $\sum_{t=1}^{T_0} \frac{2}{\log^{\eta} t} = C_{T_0}$, a constant.

Labelers with different types of tasks: We now discuss an extension where labelers' difference in their quality in labeling varies according to different types of data samples/tasks. For example, some are more proficient with labeling image data while some may be better at annotating audio data. In this case we can use contextual information to capture these differences, where a specific context refers to a different data/task type. There are two cases of interest from a technical point of view: when the space of all context information is finite, and when this space is infinite. We will denote a specific context by w and the set of all contexts as $\mathscr{W}$.

In the case of discrete context information, $|\mathscr{W}| < \infty$ and we can apply the same algorithm to learn, for each combination $\{i, w\}_{i \in \mathscr{M}, w \in \mathscr{W}}$, the pairwise labeler-context accuracy. This extension is rather straightforward except for a longer exploration phase. In fact, since exploration is needed for each labeler i under each possible context w, we may expect the regret to be $|\mathscr{W}|$ times larger compared to the previous $R(T)$. This indeed can be more precisely established using the same methodology.

The case of continuous context information is more challenging, but can be dealt with using the technique introduced in [2] for bandit problems with a continuum of arms. The main idea is to divide the infinite context information space into a finite but increasing number of subsets. For instance, if we model the context information space as $\mathscr{W} = [0,1]$ then we can divide this unit interval into $v(t)$ sub-intervals: $[0, \frac{1}{v(t)}], ..., [\frac{v(t)-1}{v(t)}, 1]$, with $v(t)$ being an increasing sequence w.r.t. t. Denote these intervals as $B_i(t)$, $i = 1, 2..., v(t)$, which become more and more fine-grained with increasing t and increasing $v(t)$.

Given these intervals the learning algorithm works as follows. At time t, for each interval $B_i(t)$ we compute the estimated optimal set of labelers by calculating the estimated utility of all subsets of labelers, and this is done over the entire interval $B_i(t)$ (contexts within $B_i(t)$ are viewed as a bundle). If at time t we have context $w_i \in B_i(t)$ then this estimated optimal set is used. The regret of this procedure consists of two parts. The first part is due to selecting a sub-optimal set of labelers for $B_i(t)$ (owing to incorrect estimates of the labelers' accuracy). This part of the regret is bounded by $O(1/t^2)$. The second part of the regret arises from the fact that even if we compute the correct optimal set for interval $B_i(t)$, it may not be optimal for the specific context $w_t \in B_i(t)$. However, when $B_i(t)$ becomes sufficiently small, and under a uniform Lipschitz condition we can bound this regret as well.

Taken together, if we revise the condition for entering the exploration phase (constants $D_1(t)$ and $D_2(t)$) to grow on the order of $O(t^z \log t)$ instead of $\log t$, for some constant $0 < z < 1$, then the regret $R(T)$ in this case is on the order of $T^z \log T$; thus it remains sub-linear and therefore has a zero average regret, but this is worse than the log bound we can obtain in other cases.

We omit all technical details since they are rather direct extensions combining our previously derived results with the literatures on continuous arms.

IV. Weighted Majority Voting and its regret

The labeling performance could be further improved by employing more sophisticated majority voting mechanism. Specifically, under our online learning algorithm LS_OL, statistics over each labeler's expertise could be collected with significant confidence; this enables a weighted majority voting mechanism. In this section we analyze the regret of a similar learning algorithm using weighted majority voting.

A. Weighted Majority Voting

We start with defining the weights. At time t, after observing labels produced by the labelers, we can optimally (*a posteriori*) determine the mostly likely label of the task by solving the following:

$$\text{argmax}_{l \in \{0,1\}} P(L^*(t) = l | L_1(t), ..., L_M(t)) \ . \qquad (10)$$

Suppose at time t the true label for task k is 1. Then we have,

$$\begin{aligned} P(L^*(t) = 1 | L_1(t), ..., L_M(t)) &= \frac{P(L_1(t), ..., L_M(t), L^*(t) = 1)}{P((L_1(t), ..., L_M(t))} \\ &= \frac{P(L_1(t), ..., L_M(t) | L^*(t) = 1) \cdot P(L^*(t) = 1)}{P((L_1(t), ..., L_M(t))} \\ &= \frac{P(L^*(t) = 1)}{P((L_1(t), ..., L_M(t))} \cdot \prod_{i: L_i(t) = 1} p_i \cdot \prod_{i: L_i(t) = 0} (1 - p_i) \ . \end{aligned}$$

And similarly we have

$$\begin{aligned} &P(L^*(t) = 0 | L_1(t), ..., L_M(t)) \\ &= \frac{P(L^*(t) = 0)}{P((L_1(t), ..., L_M(t))} \cdot \prod_{i: L_i(t) = 0} p_i \cdot \prod_{i: L_i(t) = 1} (1 - p_i) \ . \end{aligned}$$

Following standard hypothesis testing procedure and assuming equal priors $P(L^*(t)=1)=P(L^*(t)=0)$, a true label of 1 can be correctly produced if

$$\prod_{i:L_i(t)=1} p_i \cdot \prod_{i:L_i(t)=0} (1-p_i) > \prod_{i:L_i(t)=0} p_i \cdot \prod_{i:L_i(t)=1} (1-p_i) \ .$$

with ties broken randomly and equally likely. Take $\log(\cdot)$ on both sides and the above condition reduces to

$$\sum_{i:L_i(t)=1} \log \frac{p_i}{1-p_i} > \sum_{j:L_j(t)=0} \log \frac{p_j}{1-p_j} \ .$$

Indeed if $p_1=...=p_M$ the above reduces to $|\{i:L_i(t)=1\}| > |\{i:L_i(t)=0\}|$ which is exactly the simple majority voting. Under the weighted majority voting, each labeler i's decision is modulated by weight $\log \frac{p_i}{1-p_i}$. When $p_i > 0.5$, the weight $\log \frac{p_i}{1-p_i} > 0$, which may be viewed as an opinion that adds value; when $p_i < 0.5$, the weight $\log \frac{p_i}{1-p_i} < 0$, an opinion that actually hurts; when $p_i = 0.5$ the weight is zero, an opinion that does not count as it mounts to a random guess. The above constitutes the weighted majority voting rule we shall use in a revised learning algorithm and the regret analysis that follow.

Before proceeding to the regret analysis, we again first characterize the optimal labeler set selection assuming known labelers' accuracy. In this case the odd-number selection property no longer holds, but thanks to the monotonicity of $\log \frac{p_i}{1-p_i}$ in p_i we have the same monotonicity property in the optimal set and a linear-complexity solution space, when $\eta=0$ or $\eta>0$ & $c_i \equiv c$.

Theorem 4: Under the weighted majority voting and assuming $p_i \geq 0.5, \forall i$, the optimal set S_η^* is monotonic, i.e., if we have $i \in S_\eta^*$ and $j \notin S_\eta^*$ then we must have $p_i > p_j$.

The assumption that all $p_i \geq 0.5$ is for simplicity in presentation without losing generality. This is because a labeler with $p_i < 0.5$ is equivalent to another with $p_i := 1-p_i$ by flipping its label (assuming the average labeling quality is higher than 0.5). The above result is trivial when the objective consists purely of labeling accuracy ($\eta=0$): all workers are selected. In the case of $\eta>0$, the implication is more complex. On one hand, the above lemma can significantly reduce the search complexity in this case. On the other hand, in order to select and compute $\pi(S)$ (to then compute $U_\eta(S)$), determining the majority winning set is non-trivial; here the majority winning set refers to a subset $S_{\text{win}} \subseteq S$ such that $\sum_{i\in S_{\text{win}}} \frac{p_i}{1-p_i} \geq \sum_{j\in S\setminus S_{\text{win}}} \frac{p_j}{1-p_j}$. This requires another level of search. Different from simple majority where the computation is over all subsets of size $\lceil (|S|+1)/2 \rceil$ (winning is entirely determined by size), here the search can be more exhaustive. A simple heuristic can help reduce the complexity: first order all labelers in each selected S (to compute $U_\eta(S)$) in descending order of their p_i, then for each possible winning set size $1,2,...,|S|$, determine the majority winning set by sequentially adding labelers from the top to bottom, and stop when including a labeler of a rank leads to a non-winning set.

B. Main results

We now analyze the performance of a similar learning algorithm using weighted majority voting. The algorithm LS_OL is modified as follows. Denote by $W(S)=\sum_{i\in S}\log\frac{p_i}{1-p_i}$, $\forall S \subseteq \mathscr{M}$, and $\tilde{W}$ its estimated version when using estimated accuracies $\tilde{p}_i$. Denote by

$$\delta_{\min}^W = \min_{S\neq S', W(S)\neq W(S')} |W(S)-W(S')|$$

and let $\varepsilon_\eta :< \min\{\varepsilon_\eta, \delta_{\min}^W/2\}$[6], where the ε_η on the RHS is similarly defined as in simple majority voting case. At time t (suppose at exploitation phase), the algorithm selects the estimated optimal set S_t. These labelers then return their labels that divide them into two subsets, say S (with one label) and its complement $S_t\setminus S$ (with the other label). If $\tilde{W}(S) \geq \tilde{W}(S_t\setminus S)+\varepsilon_\eta$, we will call S the majority set and take its label as the voting outcome. If $|\tilde{W}(S)-\tilde{W}(S_t\setminus S)| < \varepsilon_\eta$, we will call them equal sets and randomly select one of the labels as the voting outcome. Intuitively ε_η serves as a tolerance that helps to remove the error due to inaccurate estimations. In addition, the constant $D_1(t)$ is revised to the follow:

$$D_1(t) = \log t / (\frac{1}{\max_m \max\{4C\cdot m, m\cdot n(S^m)\}} - \alpha)^2 \cdot \varepsilon_\eta^2,$$

where C is a constant satisfying

$$C > \max_i \max\{\frac{1+\varepsilon/4}{p_i}, \frac{1-\varepsilon/4}{1-p_i}, \frac{\varepsilon/4}{p_i}, \frac{\varepsilon/4}{1-p_i}\} \ .$$

With above modifications in mind, we omit the detailed algorithm description for a concise presentation. We have the following theorem on the regret of this revised algorithm (again $R_{\mathscr{C}}(T)$ possesses a similar format we omit its detail), when $\eta=0$ or $\eta>0$ & $c_i \equiv c$.

Theorem 5: The regret under weighted majority voting can be bounded uniformly in time:

$$R_\eta(T) \leq \frac{U_\eta(S_\eta^*)\cdot \log^2 T}{(\frac{1}{\max_m \max\{4C\cdot m, m\cdot n(S^m)\}} - \alpha)^2 \cdot \varepsilon_\eta^2 \cdot (a_{\min}-0.5)^2}$$
$$+ \Delta_{\max,\eta}(2\cdot \sum_{m=1}^{M} m\cdot n(S^m) + M + \frac{M^2}{2}) \cdot (2\beta_2 + \frac{1}{\alpha\cdot\varepsilon_\eta}\beta_{2-z}) \ .$$

Again the regret is on the order of $O(\log^2 T)$ in time. It has a potentially larger constant compared to that under simple majority voting[7]. However, the weighted majority voting has a better optimal solution, i.e., we are converging slightly slower to a however better target.

The proof of this theorem is omitted for brevity and because most of it parallels with the case of simple majority voting. There is however a main difference: under the weighted majority voting there is additional error in computing the weighted majority vote. Whereas under simple majority we simply find the majority set by counting the number of votes, under weighted majority the calculation of the majority set is dependent on the estimated weights $\log\frac{\tilde{p}_i}{1-\tilde{p}_i}$ which inherits errors in $\{\tilde{p}_i\}$. This additional error, in particular associated with bounding the error of getting

$$\tilde{W}(\hat{S}) - \tilde{W}(S\setminus\hat{S}) < \varepsilon_\eta, \text{ when } W(\hat{S}) > W(S/\hat{S})$$
$$\tilde{W}(\hat{S}) - \tilde{W}(S\setminus\hat{S}) \geq \varepsilon_\eta, \text{ when } W(\hat{S}) = W(S\setminus\hat{S}),$$

[6]We use $:<$ to define a smaller term than the RHS.

[7]This argument is based on comparing the upper bounds. A numerical comparison of the convergence is provided in Section VI.

for set $\hat{S} \subseteq S \subseteq \mathscr{M}$, could be separately bounded using similar methods as shown in the simple majority voting case (bounding estimation error with large enough number of samples) and can again be factored into the overall bound. This is summarized in the following lemma.

Lemma 5: At time t, $\forall \hat{S} \subseteq S \subseteq \mathscr{M}$ and its complement $S \backslash \hat{S}$, if $W(\hat{S}) > W(S \backslash \hat{S})$, then $\forall t$ at exploitation phases $\forall\ 0 < z < 1$,

$$P(\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S}) < \varepsilon_\eta) \leq |S| \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon_\eta \cdot t^{2-z}}) \ .$$

Moreover, if $W(\hat{S}) = W(S \backslash \hat{S})$

$$P(|\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S})| > \varepsilon_\eta) \leq |S| \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon_\eta \cdot t^{2-z}}) \ .$$

V. A LOWER BOUND ON THE REGRET

In this section we establish a lower bound on the regret of our online labeler selection problem. We present the results for simple majority voting with $\eta = 0$.[8]

A. $O(1)$ reassignment leads to unbounded regret

We first show that a constant number of re-assignments will lead to unbounded regret. Below we establish this by contradiction. Recall that we have used $D_2(t)$ to determine when reassignment is made, and in our algorithm we have used $D_2(t) = O(\log t)$. Suppose instead, $D_2(t)$ is given by T_0, a bounded constant. Let's consider one task with label $\theta \in \{0,1\}$. Denote the test outcomes by $x(1),...,x(T_0)$ (as given by the simple majority voting from all labelers). There are two hypotheses based on $x(\tau), \tau = 1,...,T_0$:

$$H_0 : \text{The label is 1, } \theta = 1; \ H_1 : \text{The label is 0, } \theta = 0.$$

Denote by $I(\theta_1, \theta_0)$ the Kullback-Leibler (KL) divergence between two distributions $f_X(x; \theta = 1)$ and $f_X(x; \theta = 0)$, where $f_X(x; \theta)$ denotes the sample distribution with parameter θ, which in our case is the ground-truth label. Denote the vector $[1,...,t]$ by $[t]$. The next lemma is a well established result:

Lemma 6 (Theorem 2.2, Tsybakov [27], 2009): The error probability P_e of the above hypothesis test up to time t is lower-bounded by $P_e \geq e^{-I(P_{H_0}^{[t]}, P_{H_1}^{[t]})}/2$.
Note that in our case

$$I(P_{H_0}^{[t]}, P_{H_1}^{[t]}) = E_{H_0}\left[\log \frac{f_X(x(1); \theta = 1)}{f_X(x(1); \theta = 0)} \cdots \frac{f_X(x(T_0); \theta = 1)}{f_X(x(T_0); \theta = 0)}\right]$$
$$= E_{H_0}\left[\sum_{t=1}^{T_0} \log \frac{f_X(x(1); \theta = 1)}{f_X(x(1); \theta = 0)}\right] = I(\theta_1, \theta_0) T_0 \ .$$

Since T_0 is bounded from above, $P_e > 0$, meaning that there is always a positive probability of making the wrong labeling decision. What this further means is that one can always find problem parameters whereby an algorithm will reach incorrect estimates of labelers' qualities which leads to "permanently" sub-optimal selection of labelers, resulting in unbounded regret. This is demonstrated using a counter example shown below.

[8] Again the analysis for $\eta > 0$ & $c_i \equiv c$ can be done similarly.

Suppose we have three labelers with labeling qualities being $p_1 = \frac{1}{2} + \delta, p_2 = \frac{1}{2} + \xi, p_3 = \frac{1}{2} + \xi - o(1)$, where $o(1)$ is an arbitrarily small quantity, and δ, ξ satisfies $0 < \xi < \delta < \frac{1}{2}$. This setting satisfies the assumptions we made throughout the paper: (1) $p_1 > p_2 > p_3$. (2) $0 < p_i < 1, i = 1,2,3$. (3) $\frac{p_1+p_2+p_3}{3} > 1/2$. Moreover the labeling accuracy using simple majority voting is as follows:

$$\pi(p_1, p_2, p_3) \geq \frac{1}{2} + \frac{\delta}{2} + \xi - 2\delta\xi^2 > 1/2.$$

For this 3-labeler problem we have the following proposition.

Proposition 6: With $P_e > 0$, we can always find a δ such that in the above example, the regret of any online learning algorithm is on the order of $O(T)$.

a) $D_2(t) > O(1)$: : Now consider the case with $D_2(t) > O(1)$. Using Chernoff bound we know the following holds

$$P(\frac{\sum_{\tau=1}^{D_2(t)} x(\tau)}{D_2(t)} > 1/2) = 1 - P(\frac{\sum_{\tau=1}^{D_2(t)} x(\tau)}{D_2(t)} \leq 1/2)$$
$$= 1 - P(\frac{\sum_{\tau=1}^{D_2(t)} x(\tau)}{D_2(t)} - \bar{p} \leq 1/2 - \bar{p}) \geq 1 - e^{2(\bar{p}-1/2)^2 D_2(t)} \ .$$

Notice we have used $\bar{p}$ to denote the average labeling accuracy, which is strictly larger than $1/2$. Thus $P_e \leq e^{2(\bar{p}-1/2)^2 D_2(t)} \to 0$. We have the following proposition.

Proposition 7: With $D_2(t) > O(1)$ we have

$$R(T) \geq O(\frac{\log T \cdot D_2(t)}{I(p_1, p_2) - \frac{C_1}{(C_2+\delta)^2}\delta_e}) \ , \tag{11}$$

where $C_1, C_2 > 0$ are constants and $\delta_e = \frac{P_e}{1-P_e}$.
Also from above results we see when $D_2(t) = O(1)$, it cannot be guaranteed that $I(p_1, p_2) - \frac{C_1}{(C_2+\delta_e)^2}\delta_e > 0$, under which case the bound becomes meaningless. This is another implication why we need $D_2(t) > O(1)$.

B. A refined upper bound to match

We refine our algorithm and relax the requirement on setting $D_2(t) := O(\log t)$ to any $D_2(t) > O(1)$. We have the following results to match this lower bound. In particular we prove the following results.

Theorem 8: We can refine the upper bound of LS_OL to the following. $R(T) \leq O(\log T D_2(T))$.

Tightness of $O(\log^2 T)$ for a type of policies: Note we previously had a $O(\log^2 T)$ upper bound. We show this bound is tight for a certain category of policies. We first define polynomially converging policy for our labeler selection problem.

Definition 9: A polynomially converging policy is a policy that there exists a $z > 0$ such that P_e is decreasing polynomially $P_e \leq O(t^{-z})$.

For polynomially converging policy, intuitively we need $\exp(-2(\bar{p} - 1/2)^2 D_2(t)) = O(t^{-z})$, and again following Lemma 6, we could successfully show that the number of re-assignments needs to satisfy that $D_2(t) \geq \log t$. Then $O(\log t^2)$ is tight for polynomially decreasing policies.

VI. EXPERIMENT RESULTS

In this section we validate the proposed algorithms with a few examples using both simulated and real data.

A. Simulation study

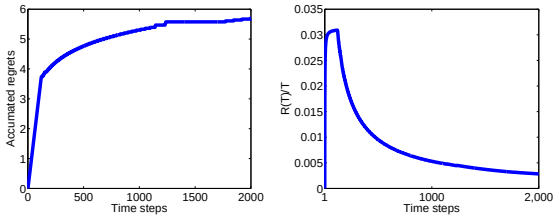
Our first setup consists of $M = 5$ labelers, whose quality $\{p_i\}$ are randomly and uniformly generated to satisfy a preset $a_{\min}$ as follows: select $\{p_i\}$ randomly between $[a_{\min}, 1]$. Note that this is a simplification because not all $\{p_i\}$ need to be above $a_{\min}$ for the requirement to hold. An example of these are shown in Table I (for $a_{\min} = 0.6$) but remain unknown to the labelers. A task arrives at each time t. We assume a unit

	L_1	L_2	L_3	L_4	L_5
p_i	0.763	0.781	0.625	0.783	0.727

TABLE I: Sample of simulation setup

labeling cost $c = 0.02$ (lablers have the same cost), and $\eta = 1$ when a linearly combined objective is used. The experiments are run for a period of $T = 2{,}000$ time units (2,000 tasks in total). The results shown below are the average over 100 runs. Denote by G_1, G_2 the *exploration constants* concerning the two constants (in $D_1(t)$ and $D_2(t)$) that control the exploration part of the learning. G_1, G_2 are set to be sufficiently large based on the other parameters:

$$(G_1, G_2) = \left(\frac{1}{\left(\frac{1}{\max_{m:m \text{ odd}} m \cdot n(S^m)} - \alpha\right)^2 \cdot \varepsilon^2}, \frac{1}{(a_{\min} - 0.5)^2}\right) .$$



(a) Accumulative regret (b) Average regret $R(T)/T$

Fig. 2: Regret of the LS_OL algorithm.

We first show the accumulative and average regret under the simple majority voting rule in Fig. 2. From the set of figures we observe a logarithmic increase of accumulated regret and correspondingly a sub-linear decrease for its average quantity. The cost regret $R_{\mathscr{C}}(T)$ has a very similar trend as mentioned earlier (recall the regret terms of $R_{\mathscr{C}}(T)$ align well with the those in $R(T)$) and is thus not show here. We then compare the performance with labeler selection to the naive crowd-sourcing algorithm, by taking a simple majority vote over the whole set of labelers each time. This is plotted in Fig. 3 in terms of the average reward at each t. There is a clear performance improvement after an initialization period (where training happens).

In addition to the logarithmic growth, we are interested in knowing how the performance is affected by the inaccuracy of the crowd expertise. These results are shown in Fig. 4. We observe the effect of different choices of $a_{\min} = 0.6, 0.7, 0.8$. To make the comparison clear, we use the average error rate defined as $\sum_{n=1}^{t} 1(S(t) \neq S^*)/t$. As expected, we see when $a_{\min}$ is small, the verification process of the labels takes more samples to become accurate. Therefore in the process more

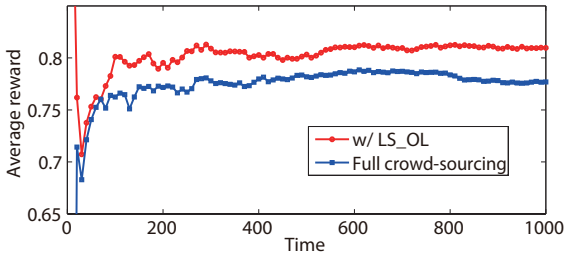


Fig. 3: Performance comparison: labeler selection v.s. full crowd-sourcing (majority voting)

error is introduced in the estimation of the labelers' qualities, which results in slower convergence.

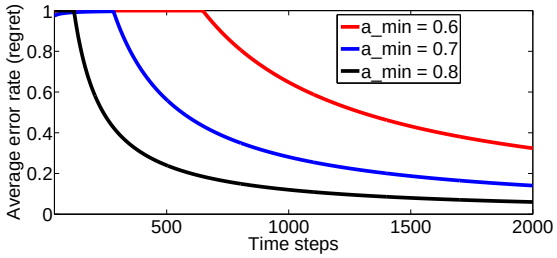


Fig. 4: Effect of $a_{\min}$: higher $a_{\min}$ leads to much better performance.

We next compare the performance between simple majority voting and weighted majority voting (both with LS_OL). One example trace of accumulated reward comparison is shown in Fig. 5; the advantage of weighted majority voting can be seen clearly. We then repeat the set of experiments and average the results over 500 runs; the comparison is shown in Table II under different number of candidate labelers (all of their labeling qualities are uniformly generated).

Despite the superior performance of weighted majority voting, the convergence (to S^*) may be slower, due to the additional error in vote counting as mentioned in Section IV. We use the linearly combined utility function defined in Eqn. (3) with $\eta = 1$, and again compare the average error rate. This comparison is shown in Fig. 6.

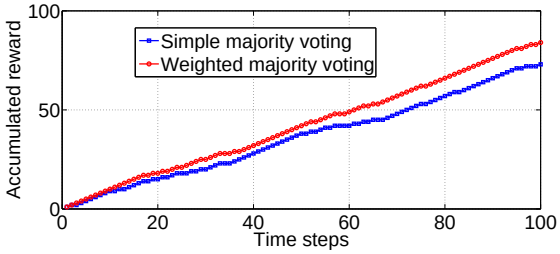


Fig. 5: Comparing weighted and simple majority voting within LS_OL: accumulated reward. $\eta = 1$.

B. Study on a real AMT dataset

We also apply our algorithm to a dataset shared at [1]. This dataset contains 1,000 images each labeled by the same set of

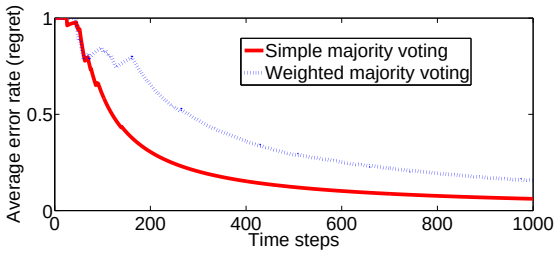


Fig. 6: Comparing weighted and simple majority voting within LS_OL: convergence of normalized regret. $\eta = 1$.

Average reward/ $M :=$	5	10	15	20
Full crowd-sourcing (majority voting)	0.5154	0.5686	0.7000	0.7997
Simple majority voting w/ LS_OL	0.8320	0.9186	0.9434	0.9820
Weighted majority voting w/ LS_OL	**0.8726**	**0.9393**	**0.9641**	**0.9890**

TABLE II: Performance comparison. There is a clear gap between crowd-sourcing results with and without using LS_OL.

5 AMTs. The labels are on the scale from 0 to 4 indicating how many scenes are seen from each image, such as filed, airport, animal, etc. A label of 0 implies no scene can be discerned. Besides the ratings from the AMTs, there is a second dataset from [1] summarizing keywords for scenes of each image. We also analyze this second dataset and count the number of unique descriptors for each image and use this count as the ground-truth or gold standard, to which the results from AMT are compared.

We start with showing the number of disagreements each AMT has with the group over the 1000 images. The total numbers of disagreement of the 5 AMTs are shown in Table III, while Fig. 7 shows the cumulative disagreement over the set of images ordered by their numerical indices in the database. It is quite clear that AMT 5 shows significant and consistent disagreement with the rest. AMT 3 comes next while AMTs 1, 2, and 4 are clearly more in general agreement.

	AMT1	AMT2	AMT3	AMT4	AMT5
# of disagree	348	353	376	338	441

TABLE III: Total number of disagreement each AMT has

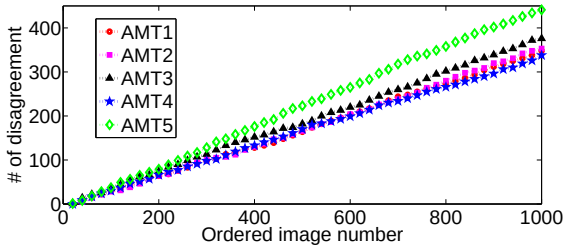


Fig. 7: Cumulated number of disagreements.

The images are not in sequential order, as the original experiment was not done in an online fashion. To test our algorithm, we will continue to use their numerical indices to order them as if they arrived sequentially in time and feed them into our algorithm. By doing so we essentially test the performance of conducting this type of labeling tasks online whereby the administrator of the tasks can dynamically alter task assignments to obtain better results. In this experiment we use LS_OL with simple majority voting and weighted majority voting, respectively, and with the addition of the detection and filtering procedure discussed in Section III-E, which is specified to eliminate the worst labeler after a certain number of steps such that the error in the rank ordering is less than 0.1. The algorithm otherwise runs as described earlier. Indeed we see this happen around step 90, as highlighted in Fig. 8 along with a comparison to using the full crowd-sourcing method with majority voting.

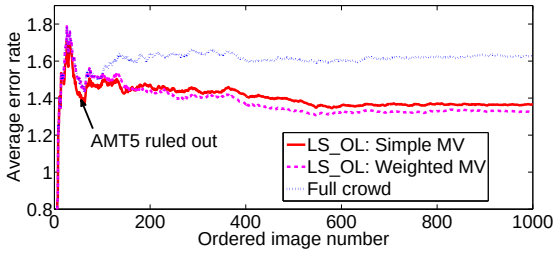


Fig. 8: Performance comparison: an online view. $\eta = 0$.

The algorithm also eventually correctly estimates the best set to consist of AMTs 1, 2, and 4. The average error (compare to the best combination in hindsight) in selecting the labelers is shown in Fig. 9[9].

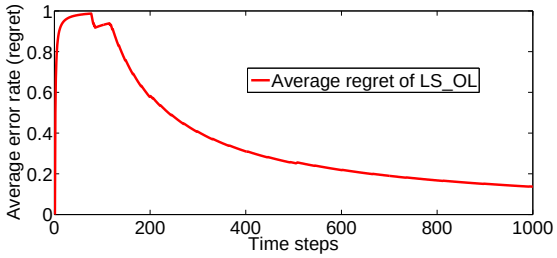


Fig. 9: Average error in labeler selection

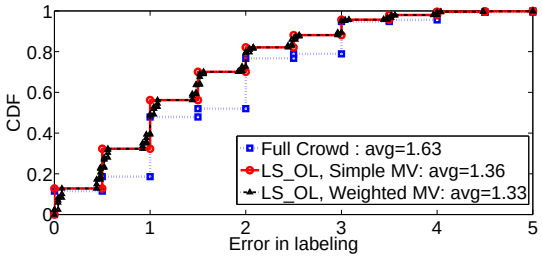


Fig. 10: Performance comparison: a summery. $\eta = 0$.

All images' labeling error as compared to the ground truth at the end of this process is shown as a CDF (error distribution over the images) in Fig. 10; note the errors are discrete due

[9]For our AMT studies, we focus on labeling accuracy so we set $\eta = 0$ in our objective. As we mentioned earlier, when $\eta = 0$ the optimal set of labelers under WMV is the full set. So there is no regret in labeler selection for WMV.

to the discrete labels. It is also worth noting that under our algorithm the cost is much lower because AMT 5 was quickly eliminated, while AMT 4 was only used very infrequently once the optimal set has been accurately inferred. The difference between simple majority voting and weighted majority voting is very marginal. This is mainly due to the fact labeler 1,2,4's performances (accuracy level) are similar to each other, which leads to similar weights, and thus similar predictions.

VII. Future works

Currently we are working on the case when there is partial and delayed feedback (ground-truth) for each task. This is different from the setting in the current paper as our inference on labeling quality is purely based on the crowd-sourced results, without any additional information. This may be made true by observing recent advances in active learning research that the quality of labels can be inferred from followed-by machine learning tasks. A preliminary analysis indicates with such assumption the regret order can be brought back to $O(\log T)$ instead of being $O(\log T^2)$.

Another immediate extension of our works is to consider the labeling quality control problem in adversarial settings where labelers' outcomes form non-stochastic bandits, compared to the current stochastic setting. We feel the non-stochastic model may be a better justification for human behaviors.

VIII. Conclusion

To our best knowledge, this is the first work formalizing and addressing the issue of labeler quality in an online fashion for the crowd-sourcing problem and proposing solutions with performance guarantee. We developed and analyzed an online learning algorithm that can differentiate between high and low quality labelers over time and select the best set for labeling tasks with $O(\log T \cdot D_2(T))$ regret uniform in time, where $D_2(T)$ is an arbitrary function with $D_2(T) > O(1)$. We also provided an order-matching lower bound. In addition, we showed how performance could be further improved by utilizing more sophisticated voting techniques. We discussed the applicability of our algorithm to more general cases when labelers' quality varies with contextually different tasks and discuss how to detect and remove malicious labelers when there is a lack of ground-truth. We validated our results via both synthetic and real world AMT data, alongside numerous observations and discussions.

Acknowledgment

This work is partially supported by the NSF under grant CNS 1422211 and DHS under grant HSHQDC-13-C-B0015.

References

[1] AMT dataset. http://tamaraberg.com/importanceDataset/.

[2] Agrawal, R. The Continuum-Armed Bandit Problem. *SIAM journal on control and optimization 33*, 6 (1995), 1926–1951.

[3] Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn. 47* (May 2002), 235–256.

[4] Chandramouli, S. S. Multi armed bandit problem: some insights.

[5] Chapelle, O., Sindhwani, V., and Keerthi, S. S. Optimization Techniques for Semi-supervised Support Vector Machines. *The Journal of Machine Learning Research 9* (2008), 203–233.

[6] Choffnes, D. R., Bustamante, F. E., and Ge, Z. Crowdsourcing Service-level Network Event Monitoring. *SIGCOMM Comput. Commun. Rev. 40*, 4 (Aug. 2010), 387–398.

[7] Dasgupta, A., and Ghosh, A. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd international conference on World Wide Web* (2013), ACM, pp. 319–330.

[8] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database.

[9] Ghosh, A., Kale, S., and McAfee, P. Who moderates the moderators?: Crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM Conference on Electronic Commerce* (New York, NY, USA, 2011), EC '11, ACM, pp. 167–176.

[10] Haklay, M., and Weber, P. Openstreetmap: User-generated Street Maps. *Pervasive Computing, IEEE 7*, 4 (2008), 12–18.

[11] Ho, C.-J., and Vaughan, J. W. Online task assignment in crowdsourcing markets. In *AAAI'12* (2012), pp. –1–1.

[12] Hua, G., Long, C., Yang, M., and Gao, Y. Collaborative Active Learning of a Kernel Machine Ensemble for Recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on* (2013), IEEE, pp. 1209–1216.

[13] Karger, D. R., Oh, S., and Shah, D. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems* (2011), pp. 1953–1961.

[14] Karger, D. R., Oh, S., and Shah, D. Efficient crowdsourcing for multi-class labeling. In *ACM SIGMETRICS Performance Evaluation Review* (2013), vol. 41, ACM, pp. 81–92.

[15] Kulis, B., Basu, S., Dhillon, I., and Mooney, R. Semi-supervised Graph Clustering: a Kernel Approach. *Machine learning 74*, 1 (2009), 1–22.

[16] Lai, T. L., and Robbins, H. Asymptotically Efficient Adaptive Allocation Rules. *Advances in Applied Mathematics 6* (1985), 4–22.

[17] Liu, Y., and Liu, M. Group Learning and Opinion Diffusion in a Broadcast Network. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on* (Oct 2013), pp. 1509–1516.

[18] Liu, Y., and Liu, M. An online learning approach to improving the quality of crowd-sourcing. In *ACM SIGMETRICS Performance Evaluation Review* (2015), vol. 43, ACM, pp. 217–230.

[19] Long, C., Hua, G., and Kapoor, A. Active Visual Recognition with Expertise Estimation in Crowdsourcing. In *Computer Vision (ICCV), 2013 IEEE International Conference on* (2013), IEEE, pp. 3000–3007.

[20] Massoulié, L., Ohannessian, M. I., and Proutière, A. Greedy-Bayes for Targeted News Dissemination. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (New York, NY, USA, 2015), SIGMETRICS '15, ACM, pp. 285–296.

[21] Natarajan, N., Dhillon, I., Ravikumar, P., and Tewari, A. Learning with Noisy Labels. In *Advances in Neural Information Processing Systems* (2013), pp. 1196–1204.

[22] Prelec, D. A bayesian truth serum for subjective data. *science 306*, 5695 (2004), 462–466.

[23] Rea, L. M., and Parker, R. A. *Designing and conducting survey research: A comprehensive guide*. John Wiley & Sons, 2012.

[24] Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. LabelMe: A Database and Web-Based Tool for Image Annotation. *Int. J. Comput. Vision 77*, 1-3 (May 2008), 157–173.

[25] Sheng, V. S., Provost, F., and Ipeirotis, P. G. Get Another Label? Improving Data Quality and Data Mining Using Multiple, Noisy Labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), ACM, pp. 614–622.

[26] Tekin, C., and Liu, M. Online Learning of Rested and Restless Bandits. *Information Theory, IEEE Transactions on 58*, 8 (2012), 5588–5611.

[27] Tsybakov, A. B. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[28] Witkowski, J., Bachrach, Y., Key, P., and Parkes, D. C. Dwelling on the negative: Incentivizing effort in peer prediction. In *First AAAI Conference on Human Computation and Crowdsourcing* (2013).

[29] Zhong, E., Fan, W., and Yang, Q. Contextual collaborative filtering via hierarchical matrix factorization. In *SDM'12* (2012), pp. 744–755.

[30] Zhu, X., and Goldberg, A. B. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning.

PROOF OF LEMMA 3

Firstly notice via union bound we have $\forall t$:

$$E[\mathscr{E}_1(t)] \leq \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} P(\tilde{U}(S^m) \geq \tilde{U}(S^*)) \ . \tag{12}$$

Now consider each term in the above summation $P(\tilde{U}(S^m) \geq \tilde{U}(S^*))$. We will use the following fact to bound it.

Lemma 7: The probability of using a sub-optimal selection S^m is bounded as follows,

$$\begin{aligned} P(\tilde{U}(S^m) \geq & \tilde{U}(S^*)) \leq P(\tilde{U}(S^m) > U(S^m) + \varepsilon) \\ & + P(\tilde{U}(S^*) < U(S^*) - \varepsilon) \ , \end{aligned} \tag{13}$$

and for $S \in \{S^m, S^*\}$ we have

$$P(|\tilde{U}(S) - U(S)| > \varepsilon) \leq n(S) \cdot \sum_{i \in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|}) \ .$$

We shall use the above lemma; its own proof is given in [18].

Consider each term $P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|})$ in the lemma

$$\begin{aligned} & P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|}) \\ & = \underbrace{P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|} \mid \frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} \leq \frac{\alpha \cdot \varepsilon}{t^z})}_{\textbf{Term 1}} \\ & \cdot P(\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} \leq \frac{\alpha \cdot \varepsilon}{t^z}) \\ & + P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|} \mid \frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} > \frac{\alpha \cdot \varepsilon}{t^z}) \\ & \cdot \underbrace{P(\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} > \frac{\alpha \cdot \varepsilon}{t^z})}_{\textbf{Term 2}} \ , \end{aligned} \tag{14}$$

where $0 < z < 1$ is a constant. This is different from the classical learning problem in the sense we need to deal with extra errors associated with imperfect feedbacks. The first term takes care of the event when the sum of error is lower than certain threshold while the second term captures the other case.

For **Term 1** the conditional probability is bounded as:

$$\begin{aligned} & P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|} \mid \frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} \leq \frac{\alpha \cdot \varepsilon}{t^z}) \\ & \leq P(|\tilde{p}_i - p_i| > (\frac{1}{n(S) \cdot |S|} - \frac{\alpha}{t^z}) \cdot \varepsilon) \\ & \leq 2 \cdot e^{-2((\frac{1}{n(S)\cdot|S|} - \frac{\alpha}{t^z})\cdot\varepsilon)^2 \cdot D_1(t)} \leq \frac{2}{t^2} \ , \end{aligned} \tag{15}$$

since $D_1(t) = \frac{1}{(\frac{1}{n(S)\cdot|S|} - \alpha)^2 \cdot \varepsilon^2} \cdot \log t$. Consider **Term 2**,

$$\begin{aligned} & P(\frac{\sum_{k:k\in E(t)} 1\{y_k^* = 0\}}{|E(t)|} > \frac{\alpha \cdot \varepsilon}{t^z}) \\ & \leq \frac{\frac{E[\sum_{k:k\in E(t)} 1\{y_k^* = 0\}]}{|E(t)|}}{\frac{\alpha\cdot\varepsilon}{t^z}} = \frac{\frac{\sum_{k:k\in E(t)} E[1\{y_k^* = 0\}]}{|E(t)|}}{\frac{\alpha\cdot\varepsilon}{t^z}} \ , \end{aligned} \tag{16}$$

by the Markov inequality. Note more strict bound could be obtained via other bounding techniques. Consider each term in the summation

$$\begin{aligned} E[1\{y_k^* = 0\}] &= P(y_k^* = 0) = P(\sum_{n=1}^{\hat{N}_k(t)} 1\{y_k(n)\} > 0.5 \cdot \hat{N}_k(t)) \\ & \leq e^{-2(a_{\min} - 0.5)^2 \cdot \hat{N}_k(t)} \leq \frac{1}{t^2} \ , \end{aligned}$$

where $\hat{N}_k(t)$ is the number of feedbacks received for task k upto time t; the inequality is due to the fact that $\hat{N}_k(t) \geq D_2(t) \geq 1/(a_{\min} - 0.5)^2 \log t$. This means that for each labeler, it has performed on at least $D_1(T)$ tasks, and each task must have at least $D_2(T)$ testing results available.

Consequently we have

$$P(\frac{\sum_{k\in E(t)} I\{y_k^* = 0\}}{|E(t)|} > \frac{\alpha \cdot \varepsilon}{t^z}) \leq \frac{1/t^2}{\alpha \cdot \varepsilon / t^z} = \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}} \ .$$

The other two terms in the summation are bounded by 1 since they are probability measures. Summing up, we have

$$P(|\tilde{U}(S) - U(S)| > \varepsilon) \leq n(S) \cdot |S| \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) \ . \tag{17}$$

Summing over S^m, m odd completes the proof.

PROOF FOR PROPOSITION 6

Let's ignore the $o(1)$ quantity for now: as we could easily show $\pi(\cdot)$ is linear in each p_i so the $o(1)$ change in each p_i will only result in a $o(1)$ change in $\pi(\cdot)$. First

$$\pi(\frac{1}{2} + \delta_e, \frac{1}{2} + \varepsilon, \frac{1}{2} + \varepsilon) = \frac{1}{2} + \frac{\delta_e}{2} + \varepsilon - 2\delta_e\varepsilon^2 \ .$$

Compare with $1 - \delta_e$ we know we can find a (δ_e, ε) such that

$$\frac{1}{2} + \frac{\delta_e}{2} + \varepsilon - 2\delta_e\varepsilon^2 < \frac{1}{2} + \delta_e \Leftrightarrow \varepsilon < \frac{\delta_e}{2} + 2\delta_e\varepsilon^2 \ .$$

Now with error probability P_e we have the change in perception for each labelers' accuracy as follows

$$\begin{aligned} \tilde{p}_1 &= (\frac{1}{2} + \delta_e)(1 - P_e) + (\frac{1}{2} - \delta_e)P_e = \frac{1}{2} + \delta_e(1 - 2P_e) \ , \\ \tilde{p}_2 &= (\frac{1}{2} + \varepsilon)(1 - P_e) + (\frac{1}{2} - \varepsilon)P_e = \frac{1}{2} + \varepsilon(1 - 2P_e) \ , \\ \tilde{p}_3 &= (\frac{1}{2} + \varepsilon)(1 - P_e) + (\frac{1}{2} - \varepsilon)P_e = \frac{1}{2} + \varepsilon(1 - 2P_e) \ . \end{aligned}$$

First of all when $P_e > \frac{1}{2}$, we know $\tilde{p}_2 > \tilde{p}_1$, which will lead to the case that optimal set of labelers will be different from the case with p_1, p_2, p_3. When $P_e = \frac{1}{2}$, we will have $\tilde{p}_1 = \tilde{p}_2 = \tilde{p}_3 = \frac{1}{2}$. So the optimal solution does not equal to selecting labeler 1, which again leads to unbounded regrets. Now consider the case with $P_e < \frac{1}{2}$:

$$\begin{aligned} \pi(\tilde{p}_1, \tilde{p}_2, \tilde{p}_3) = & \frac{1}{2} + \frac{\delta_e(1 - 2P_e)}{2} + \delta_e(1 - 2P_e) \\ & - 2\delta_e(1 - 2P_e)(\varepsilon(1 - 2P_e))^2 \ . \end{aligned}$$

Compare it with $\frac{1}{2} + \delta_e(1 - 2P_e)$ we know

$$\begin{aligned} & \frac{1}{2} + \frac{\delta_e(1 - 2P_e)}{2} + \delta_e(1 - 2P_e) - 2\delta_e(1 - 2P_e)(\varepsilon(1 - 2P_e))^2 \\ & > \frac{1}{2} + \delta_e(1 - 2P_e) \Leftrightarrow \varepsilon > \frac{\delta_e}{2} + 2\delta_e\varepsilon^2(1 - 2P_e)^2 \ . \end{aligned}$$

Depending on different P_e we know we could choose a pair of (ε, δ_e) such that $\varepsilon < \frac{\delta_e}{2} + 2\delta_e\varepsilon^2,\ \varepsilon > \frac{\delta_e}{2} + 2\delta_e\varepsilon^2(1-2P_e)^2$, as above functions are all continuous in (ε, δ_e). So for any P_e we can find an example that based on $\tilde{p}_1, \tilde{p}_2, \tilde{p}_3$ the optimal solution set will be different from the one with p_1, p_2, p_3. Then following classical MAB results we will know the learning will converge to the sub-optimal solution which will make the learning regret being at the order of $O(T)$.

Proof for Proposition 7

Now at each time t consider the hypothesis testing on whether a sub-optimal labeler is better than an optimal one, based on collected samples. Upto time t the number of making a wrong decision for above hypothesis is lower bounded by the summation of the event when a wrong ordering of the labelers occurs; as in cases with the top labeler being the optimal selection, a wrong ordering leads to a wrong selection.

Consider the following example with two hypothesis with parameters drawing from parameter space Θ. (Hypothesis H_i corresponds to parameter space θ_i.) Particularly suppose

$$\theta_0 = \{p_1, p_2, p_3 : p_1 > p_2 > p_3\} ,$$
$$\theta_1 = \{p_1', p_2, p_3 : p_2 > p_1' > p_3\} .$$

That is H_0 believes $p_1 > p_2$ while H_1 represents the hypothesis $p_2 > p_1$.

Denote by $T(t)$ as the number of sub-optimal arm selection upto time t. Then we have

$$\sup_\theta E_\theta[T(t)] = \sup_\theta \sum_{\tau=1}^{t} P_\theta(S(\tau) \neq S^*)$$
$$\geq \sum_{\tau=1}^{t} \frac{P_{\theta_0}(S(\tau) \neq S^*) + P_{\theta_1}(S(\tau) \neq S^*)}{2} \geq \sum_{\tau=1}^{t} \frac{e^{-I(P_{H_0}^\tau, P_{H_1}^\tau)}}{4} .$$

Denote the observation sequence as $X_1, ..., X_t$. Now consider each term in the summation

$$I(P_{H_0}^\tau, P_{H_1}^\tau) = E_{\theta_0}\left(\log(\frac{\tilde{f}(X_1, p_1)}{\tilde{f}(X_1, p_1')}\frac{\tilde{f}(X_2, p_1)}{\tilde{f}(X_2, p_1')} \cdots \frac{\tilde{f}(X_{T(\tau)}, p_1')}{\tilde{f}(X_{T(\tau)}, p_1')})\right)$$
$$= E_{\theta_0}[\sum_{t=1}^{T(\tau)} \log\frac{\tilde{f}(X_1, p_1)}{\tilde{f}(X_1, p_1')}] = \tilde{I}(p_1, p_1')E_\theta[T(\tau)] .$$

Notice each distribution we have used $\tilde{f}$ to denote this is rather a noisy observation.

There we have (as similarly argued in [4])

$$S_t \geq \frac{1}{4} \cdot \sum_{\tau=1}^{t} e^{-\tilde{I}(p_1, p_1')E_{\theta_0}[T(\tau)]}$$
$$\geq \frac{1}{4} \cdot \sum_{\tau=1}^{t} e^{-\tilde{I}(p_1, p_1')\sup_\theta E_\theta[T(\tau)]}$$
$$= \frac{1}{4} \cdot \sum_{\tau=1}^{t} e^{-\tilde{I}(p_1, p_1')S_\tau} \geq \frac{1}{4} \cdot \sum_{\tau=1}^{t} e^{-\tilde{I}(p_1, p_1')S_t}$$
$$= \frac{t}{4} e^{-\tilde{I}(p_1, p_1')S_t} .$$

Take log on both sides and rearrange we know

$$S_t \geq \frac{\log t}{\tilde{I}(p_1, p_1')} + o(\frac{\log t}{\tilde{I}(p_1, p_1')}) . \tag{18}$$

Now consider $\sup_{P_e} S_t$ and we start with bounding $\tilde{I}(p_1, p_1')$. Consider the following fact.

$$\tilde{I}(p_1, p_1') = E_{\theta_0}(\log\frac{\tilde{f}(x, p_1)}{\tilde{f}(x, p_1')})$$
$$= E_{\theta_0}(\log\frac{f(x, p_1)(1-P_e) + (1-f(x, p_1))P_e}{f(x, p_1')(1-P_e) + (1-f(x, p_1'))P_e})$$
$$= E_{\theta_0}(\log\frac{f(x, p_1) + \frac{P_e}{1-2P_e}}{f(x, p_1') + \frac{P_e}{1-2P_e}})$$

For each $x \in \{0, 1\}$, consider each function $\log\frac{f(x,p_1)+\frac{P_e}{1-2P_e}}{f(x,p_1')+\frac{P_e}{1-2P_e}}$. Denote $\delta_e = \frac{P_e}{1-2P_e}$ and

$$g(\delta_e) = \log\frac{f(x, p_1) + \delta_e}{f(x, p_1') + \delta_e} , \delta_e \geq 0. \tag{19}$$

By checking the second order derivative we can easily show that when $f(x, p_1) \geq f(x, p_1')$, $g(\delta_e)$ is convex in δ_e while when $f(x, p_1) < f(x, p_1')$, $g(\delta_e)$ is concave. Therefore we have when $f(x, p_1) \geq f(x, p_1')$,

$$g(\delta_e) \geq g(0) + g'(0)\delta_e = \log\frac{f(x, p_1)}{f(x, p_1')} + \frac{f(x, p_1) - f(x, p_1')}{f(x, p_1)f(x, p_1')}\delta_e .$$

While when $f(x, p_1) < f(x, p_1')$,

$$g(\delta_e) \geq g(0) - g'(\delta_e)(-\delta_e)$$
$$= \log\frac{f(x, p_1)}{f(x, p_1')} + \frac{f(x, p_1) - f(x, p_1')}{(f(x, p_1) + \delta_e)(f(x, p_1') + \delta_e)}\delta_e .$$

Denote $-C_1 = \min_x f(x, p_1) - f(x, p_1')$, and $C_2 = \min_{x,\theta} f(x, \theta)$. Since we cannot have a probability measure being strictly larger than another on each outcome and for two different measures we know $C_1 > 0, C_2 > 0$. Then

$$g(\delta_e) \geq g(0) - \frac{C_1}{(C_2 + \delta_e)^2}\delta_e. \tag{20}$$

Then

$$E_{\theta_0}(\log\frac{f(x, p_1) + \frac{P_e}{1-2P_e}}{f(x, p_1') + \frac{P_e}{1-2P_e}}) \geq E_{\theta_0}(\log\frac{f(x, p_1)}{f(x, p_1')}) - \frac{C_1}{(C_2 + \delta_e)^2}\delta_e.$$

That is

$$\tilde{I}(p_1, p_1') \geq I(p_1, p_1') - \frac{C_1}{(C_2 + \delta_e)^2}\delta_e .$$

Then

$$S_t \geq \frac{\log t}{I(p_1, p_1') - \frac{C_1}{(C_2+\delta_e)^2}\delta_e} . \tag{21}$$

We now see since $\delta_e = \frac{P_e}{1-2P_e}$, when $P_e \to 0$, $\delta_e \to 0$ and so is $\frac{C_1}{(C_2+\delta_e)^2}\delta_e$.

We can easily prove that $I(p_1, p_2 - x)$ is increasing in x; so the lower bound is again be bounded by setting $x = 0$ that is by setting $p_1' = p_2$. Combine above analysis we know we can achieve a lower bound as $\log t D_2(t)$.