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

Yang Liu, Mingyan Liu
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor
Email: {youngliu,mingyan}@eecs.umich.edu

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) cannot be directly verified; 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 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 present numerical results using both simulation and a real dataset on a set of images labeled by Amazon Mechanic Turks (AMT).

Categories and Subject Descriptors

F.1.2 [**Modes of Computation**]: Online computation; I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Applications**]: Data mining

SIGMETRICS'15, June 15–19, 2015, Portland, OR, USA.

http://dx.doi.org/10.1145/2745844.2745874.

General Terms

Algorithm, Theory, Experimentation

Keywords

Crowd-sourcing, online learning, quality control

1. 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 [19,22], 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. [5,15,25] 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, advances in instrumentation and miniaturization, combined with frameworks like participatory sensing, rush in enormous quantities of unlabeled data.

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 [7] and LabelMe [21]. 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/information diffusion [9], 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, 22], 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 biases, and sometimes irresponsible behavior. At the same time, the cost in having a 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 paper 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 a multi-armed bandit (MAB) problem, see e.g., [4, 16, 23]. 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 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 majority voting rule 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 [8], 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). In [13] an iterative algorithm is proposed that infers labeling quality using a process similar to belief propagation and it is shown that label aggregation based on this method outperforms simple majority voting. Another example is the family of matrix factorization or matrix completion based methods, see e.g., [24], 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 study [14] has examined the fundamental trade-off between labeling accuracy and redundancy in task assignment in crowd-sourcing systems. In particular, it is shown 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.

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 [18], 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.

Our work differs from the above as we do not require an oracle or the availability of ground-truth; we instead impose a mild assumption on the collective quality of the crowd (without which crowd-sourcing would be useless and would not have existed). 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 broadly applicable to generic crowd-sourcing task assignments 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.

4. Our validation includes both simulation and the use of a real-world AMT dataset.

The remainder of the paper is organized as follows. We formulate our problem in Section 2. In Sections 3 and 4 we introduce our learning algorithm along with regret analysis under a simple majority and weighted majority voting rule, respectively. We extend our model to the case where labelers' expertise may be task-dependent in Section 5. Numerical experiments are presented in Section 6. Section 7 concludes the paper.

2. PROBLEM FORMULATION AND PRELIMINARIES

2.1 The crowd-sourcing model

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

1. *User*. There is a single user with a sequence of tasks (unlabeled data) to be performed/labeled. Our proposed online 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.

2. *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 $\mathscr{M} = \{1,2,...,M\}$. A labeler i produces the true label for each assigned task with probability p_i independent of the task; a more sophisticated task-dependent version is discussed in Section 5. 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 we consider non-trivial cases with $0 < p_i < 1, \forall i$. These quantities are unknown to the user a priori. We will also assume that the accuracy of the collection of labelers satisfies $\bar{p} := \sum_{i=1}^{M} \frac{p_i}{M} > \frac{1}{2}$, and that $M > \frac{\log 2}{2(\bar{p}-1/2)^2}$. The justification and implication of these assumptions are discussed in more detail in Section 2.3.

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[1]. 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 first 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 label output:

$$L^*(t) = \text{argmax}_{l \in \{0,1\}} \sum_{i \in S_t} I_{L_i(t)=l} \;, \tag{1}$$

with ties (i.e., $\sum_{i \in S_t} I_{L_i(t)=0} = \sum_{i \in S_t} I_{L_i(t)=1}$, where I denotes the indicator function) broken randomly.

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

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

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

$$\mathscr{C}(S) = \sum_{i \in S} c_i, \; S \subseteq \mathscr{M} \;. \tag{3}$$

Extensions of our analysis to other forms of cost functions are feasible though with more cumbersome notations. Denote

$$S^* = \text{argmax}_{S \subseteq \mathscr{M}} \, \pi(S) \;, \tag{4}$$

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. In most crowd-sourcing systems the main goal is to obtain high quality labels while the cost accrued is a secondary issue. For completeness, however, we will also analyze the tradeoff between the two. Therefore we shall adopt two objectives when designing an efficient online algorithm: choosing the best set of labelers while keeping the cost low.

It should be noted that one can also combine labeling accuracy and cost to form a single objective function, such as $\pi(S) - \mathscr{C}(S)$. The resulting analysis is quite similar to and can be easily reproduced from that presented in this paper .

2.2 Off-line 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 $\{p_i\}_{i \in \mathscr{M}}$ and the aggregation rule used to compute the final label. While there is a combinatorial number of possible selections, the next two results combined lead to a very simple and linear-complexity procedure in finding the optimal S^*.

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

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

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

2.3 The lack of ground truth

As mentioned, a key difference between our model and many other studies on crowd-sourcing 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 the standard MAB problem.

Given the lack of ground-truth, the crowd-sourcing 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 have assumed that $\bar{p} = \sum_{i=1}^{M} \frac{p_i}{M} > 0.5$, i.e., the average labeling quality is higher than 0.5 or a random guess; this is a common assumption in the crowd-sourcing literature (see e.g., [8]). 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.

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

[1] Indeed in our experiment shown later in Section 6, our algorithm is applied to a non-binary multi-label case.

correct and 0 otherwise. Using Chernoff Hoeffding's inequality we have

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

Define $a_{\min} := P(\frac{\sum_{i=1}^{M} X_i}{M} > 1/2)$; 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. It also follows that we have:

$$P(\frac{\sum_{i\in S^*} X_i}{s^*} > 1/2) \geq P(\frac{\sum_{i=1}^{M} X_i}{M} > 1/2),$$

where the inequality is due to the definition of the optimal set S^*.

3. 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.

3.1 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 design of the exploration step is complicated by the additional estimation needs 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 consistent 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 (e.g. those with multiple choices), 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., [20]. For tasks like labeling images, a labeler may be given identical images repeatedly or each time with small, added 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 entire 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-labeled). These tasks will simply be given a random label (e.g., with error probability $1/2$) but their numbers are limited by the frequency of an exploration step ($\sim \log^2 T$), as we shall detail later.

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 $y_k(1), y_k(2), \cdots, y_k(N)$:

$$y_k^*(N) = \begin{cases} 1, & \frac{\sum_{n=1}^{N} I_{y_k(n)=1}}{N} > 0.5 \\ 0, & \text{otherwise} \end{cases} , \tag{5}$$

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. A tester task is always assigned to all labelers for labeling. Following our earlier assumption that the repeated assignments of the same task use identical and independent variants of the task, we will also take the repeated outcomes $y_k(1), y_k(2), \cdots, y_k(N)$ to be independent and statistically identical.

Denote by $E(t)$ the set of tasks assigned to the M labelers during exploration steps 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:

$$\mathscr{O}(t) = I_{|E(t)| \leq D_1(t) \text{ or } \exists k \in E(t) \text{ s.t. } \hat{N}_k(t) \leq D_2(t)} ,$$

where

$$D_1(t) = \frac{1}{(\frac{1}{\max_{m:m \text{ odd}} m\cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2} \cdot \log t ,$$
$$D_2(t) = \frac{1}{(a_{\min} - 0.5)^2} \cdot \log t ,$$

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 a bounded constant, 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 when either an insufficient number of tester tasks have been assigned under exploration or some tester task has been assigned insufficient number of times in exploration.

Our online algorithm for labeler selection is formally stated in Figure 1. The LS_OL 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.

3.2 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

$$R(T) = T \cdot U(S^*) - \sum_{t=1}^{T} U(S_t) ,$$
$$R_{\mathscr{C}}(T) = \sum_{t=1}^{T} \mathscr{C}(S_t) - T \cdot \mathscr{C}(S^*) ,$$

where S_t is the selection made at time t by our algorithm; if t happens to be an exploration step then $S_t = \mathscr{M}$ and $U(S_t)$ is either 1/2 due to random guess of an arriving task that is not labeled, or $a_{\min}$ when a tester task is labeled for the first time. $R(T)$ and $R_{\mathscr{C}}(T)$ respectively capture the regret of the learning algorithm in perfor-

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 $\mathscr{O}(t) = 1$, 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 notation $I(\cdot)$):

$$y^*_k(\hat{N}_k(t)) = I\left(\frac{\sum_{\hat{t}=1}^{\hat{N}_k(t)} y_k(\hat{t})}{\hat{N}_k(t)} > 0.5\right) .$$

2.3: Update labelers' accuracy estimate $\forall i \in \mathscr{M}$:

$$\tilde{p}_i = \frac{\sum_{k \in E(t), k \text{ arrives at time } \hat{t}} I(L_i(\hat{t}) = y^*_k(\hat{N}_k(t)))}{|E(t)|} .$$

3: Else if $\mathscr{O}(t) = 0$, the algorithm exploits and computes:

$$S_t = \text{argmax}_m \tilde{U}(S^m) = \text{argmax}_{S \subseteq \mathscr{M}} \tilde{\pi}(S) ,$$

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.

Figure 1: Description of LS_OL

mance and in cost. Define:

$$\Delta_{\max} = \max_{S \neq S^*} U(S^*) - U(S),\ \delta_{\max} = \max_{i \neq j} |p_i - p_j| ,$$
$$\Delta_{\min} = \min_{S \neq S^*} U(S^*) - U(S),\ \delta_{\min} = \min_{i \neq j} |p_i - p_j| .$$

ε is a constant such that $\varepsilon < \min\{\frac{\Delta_{\min}}{2}, \frac{\delta_{\min}}{2}\}$. For analysis we assume $U(S^i) \neq U(S^j)$ if $i \neq j$[2]. Define the sequence $\{\beta_n\}$: $\beta_n = \sum_{t=1}^{\infty} \frac{1}{t^n}$. Our main theorem is stated as follows.

THEOREM 3. *The regret can be bounded uniformly in time:*

$$R(T) \leq \frac{U(S^*)}{(\frac{1}{\max_{m:m\ odd} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2 \cdot (a_{\min} - 0.5)^2} \cdot \log^2(T)$$
$$+ \Delta_{\max}(2 \sum_{\substack{m=1 \\ m\ odd}}^{M} m \cdot n(S^m) + M) \cdot (2\beta_2 + \frac{1}{\alpha \cdot \varepsilon}\beta_{2-z}) , \qquad (6)$$

$$R_{\mathscr{C}}(T) \leq \frac{\sum_{i \notin S^*} c_i}{(\frac{1}{\max_{m:m\ odd} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2} \cdot \log T$$
$$+ \frac{\sum_{i \in \mathscr{M}} c_i}{(\frac{1}{\max_{m:m\ odd} m \cdot n(S^m)} - \alpha)^2 \cdot \varepsilon^2 \cdot (a_{\min} - 0.5)^2} \cdot \log^2(T)$$
$$+ (M - |S^*|) \cdot (2 \sum_{\substack{m=1 \\ m\ odd}}^{M} m \cdot n(S^m) + M) \cdot (2\beta_2 + \frac{1}{\alpha \cdot \varepsilon}\beta_{2-z}) , \qquad (7)$$

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

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, crowd-sourcing generates ground-truth outputs with higher probability and thus 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 $O(\frac{2^m \sqrt{m}}{\sqrt{2\pi}})$.

3.3 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 appendices. There are a few that we omit for brevity; in those cases sketches are provided.

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:*

$$E[R_e(T)] \leq U(S^*) \cdot (D_1(T) \cdot D_2(T)) . \qquad (8)$$

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:*

$$E[R_x(T)] \leq \Delta_{\max}\left(E[\sum_{t=1}^{T}(\mathscr{E}_1(t) + \mathscr{E}_2(t))]\right) . \qquad (9)$$

Here $\mathscr{E}_1(t) = I_{S_t \neq S^*}$, conditioned on correct ordering of labelers, counts the event when a sub-optimal section (other than S^*) was used at time t based on the current estimates $\{\tilde{p}_i\}$. $\mathscr{E}_2(t)$ counts the event when 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 (9) separately. In this part of the analysis we only consider those times t when the algorithm exploits.

[2]This can be precisely established when $p_i \neq p_j, \forall i \neq j$.

LEMMA 3. *At time t we have:*

$$E[\mathscr{E}_1(t)] \le \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} m \cdot n(S^m) \cdot (\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) . \tag{10}$$

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:*

$$E[\mathscr{E}_2(t)] \le M(\frac{2}{t^2} + \frac{1}{\alpha \cdot \varepsilon \cdot t^{2-z}}) . \tag{11}$$

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

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

Since $\beta_{2-z} < \infty$ for $z < 1$, we have bounded the exploitation regret by a constant. Summing over all terms in $E[R_e(T)]$ and $E[R_x(T)]$ we obtain the main theorem.

3.4 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* steps the additional cost is upper-bounded by

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

Using earlier results we know the cost regret $R_{\mathscr{C}}(T)$ will be similar to $R(T)$ with both terms bounded by either a log term or a constant. Plugging 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.

3.5 Discussion

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

3.5.1 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 $\tau_k \le \tau_{\max}, \forall k$. Then following similar analysis we can show the 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.(6).

3.5.2 Prior knowledge on several constants

The second assumption concerns 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., [3,17,23]. Specifically, let

$$\varepsilon_t = \frac{1}{\log^{\eta}(t)}, \text{ for some } \eta > 0 .$$

Replace $\log(t)$ with $\log^{1+2\eta}(t)$ in $D_1(t)$ and $D_2(t)$ it can be shown that $\exists T_0 \quad \text{s.t.}: \quad \varepsilon_{T_0} < \varepsilon$. Thus the regret associated with using an imperfect ε_t is bounded by $\sum_{t=1}^{T_0} 2\varepsilon_t = \sum_{t=1}^{T_0} \frac{2}{\log^{\eta} t} = C_{T_0}$, a constant.

3.5.3 Detecting bad/incompetent labelers

The last assumption we discuss concerns the quality of the set of labelers, assumed to satisfy the condition $\min\{a_{\min}, \bar{p}\} > 0.5$. Recall the bounds were derived based on this assumption and are indeed functions of $a_{\min}$. While in this discussion we will not seek to relax this assumption, below we describe a simple "vetting" procedure that can be easily incorporated into the LS_OL algorithm to quickly detect and filter out outliner labelers so that over the remaining labelers we can achieve higher values of $a_{\min}$ and $\bar{p}$, and consequently a better bound. The procedure keeps count of the number of times a labeler differs from the majority opinion during the exploration steps, then over time we can safely eliminate those with high counts.

The justification behind this procedure is as follows. Let random variable $Z_i(t)$ denote whether labeler i agrees with the majority vote in labeling a task in a given assignment in exploration step t: $Z_i(t) = 1$ if they disagree and 0 otherwise. Then

$$\begin{aligned} P(Z_i(t) = 1) &= (1 - p_i) \cdot \pi(\mathscr{M}) + p_i \cdot (1 - \pi(\mathscr{M})) \\ &= \pi(\mathscr{M}) + p_i \cdot (1 - 2\pi(\mathscr{M})) , \end{aligned} \tag{12}$$

where recall $\pi(\mathscr{M})$ is the probability the majority vote is correct. Under the same assumption $a_{\min} > 1/2$ we have $\pi(\mathscr{M}) > 1/2$, and it follows that $P(Z_i(t) = 1)$ is decreasing in p_i, i.e., the more accurate a labeler is, the less likely she is going to disagree with the majority vote, as intuition would suggest. It further follows that for $p_i > p_j$ we have

$$\varepsilon_{ij} := E[Z_i(t) - Z_j(t)] < 0 .$$

Similarly, if we consider the disagreement counts over N assignments, $\sum_{t=1}^{N} Z_j(t)$, then for $p_i > p_j$ we have

$$\begin{aligned} &P(\sum_{t=1}^{N} Z_i(t) < \sum_{t=1}^{N} Z_j(t)) \\ &= P(\frac{\sum_{t=1}^{N}(Z_i(t) - Z_j(t))}{N} < 0) \\ &= P(\frac{\sum_{t=1}^{N}(Z_i(t) - Z_j(t))}{N} - \varepsilon_{ij} < -\varepsilon_{ij}) \\ &\ge 1 - e^{-2\varepsilon_{ij}^2 N} . \end{aligned} \tag{13}$$

That is, if the number of assignments N is on the order of $\log T/\varepsilon_{ij}^2$, then the above probability approaches 1, which bounds the likelihood that labeler i (higher quality) will have fewer number of disagreements than labeler j. Therefore if we rank and order the labelers in decreasing order of their accumulated disagreement counts then the worst labeler is going to be at the top of the list with increasing probability (approaching 1). If we eliminate the worst performer, then we improve $a_{\min}$ which leads to better bounds as shown in Eqn. (6) and Eqn. (7). Compared to the exploration steps detailed earlier where in order to differentiate labelers' expertise (by estimating p_i) $O(\log^2 T)$ assignments are needed, here we only need $O(\log T)$ assignments, a much faster process. In practice, we could decide to remove the worst labeler when the probability of not making an error (per Eqn. (13)) exceeds a certain threshold.

4. WEIGHTED MAJORITY VOTE AND ITS REGRET

The crowd-sourced labeling performance could be 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.

4.1 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:

$$\operatorname{argmax}_{l\in\{0,1\}} P(L^*(t)=l|L_1(t),...,L_M(t)) \ . \tag{14}$$

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

$$\begin{aligned}
&\prod_{i:L_i(t)=1} p_i \cdot \prod_{i:L_i(t)=0}(1-p_i) \\
&\qquad > \prod_{i:L_i(t)=0} p_i \cdot \prod_{i:L_i(t)=1}(1-p_i) \ .
\end{aligned}$$

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 rule. Under the weighted majority vote, 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 vote 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.

THEOREM 4. *Under the weighted majority vote and assuming $p_i\geq 0.5,\forall i$, the optimal set S^* is monotonic, i.e., if we have $i\in S^*$ and $j\notin S^*$ then we must have $p_i>p_j$.*

The assumption of all $p_i\geq 0.5$ is for simplicity in presentation, because a labeler with $p_i<0.5$ is equivalent to another with $p_i:=1-p_i$ by flipping its label.

4.2 Main results

We now analyze the performance of a similar learning algorithm using weighted majority vote. The algorithm LS_OL is modified as follows. Denote by

$$W(S)=\sum_{i\in S}\log\frac{p_i}{1-p_i},\ \forall S\subseteq\mathscr{M} \ , \tag{15}$$

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 $\varepsilon<\delta_{\min}^W/2$. At time t (assuming in exploitation), 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 $S_t\backslash S$ (with the other label). If $\tilde{W}(S)\geq\tilde{W}(S_t\backslash S)+\varepsilon$, we will call S the majority set and take its label as the voting outcome. If $|\tilde{W}(S)-\tilde{W}(S_t\backslash S)|<\varepsilon$, we will call them equal sets and randomly select one of the labels as the voting outcome. Intuitively ε 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:

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

where C is a constant satisfying

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

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 ($R_{\mathscr{C}}(T)$ has a very similar format and its detail is omitted).

THEOREM 5. *The regret under weighted majority vote can be bounded uniformly in time:*

$$\begin{aligned}
R(T)\leq&\frac{U(S^*)}{\left(\frac{1}{\max_m\max\{4C\cdot m,m\cdot n(S^m)\}}-\alpha\right)^2\cdot\varepsilon^2\cdot(a_{\min}-0.5)^2}\log^2 T \\
&+\Delta_{\max}\left(2\cdot\sum_{m=1}^{M}m\cdot n(S^m)+M+\frac{M^2}{2}\right)\cdot\left(2\beta_2+\frac{1}{\alpha\cdot\varepsilon}\beta_{2-z}\right) \ .
\end{aligned}$$

Again the regret is on the order of $O(\log^2 T)$ in time. It has a larger constant compared to that under simple majority vote. However, the weighted majority vote has a better optimal solution, i.e., we are converging slightly slower to a however better target. Meanwhile note that by using weighted majority vote on the testers, $a_{\min}$ can be also be potentially increased which leads to a better upper bound.

The proof of this theorem is omitted for brevity and because most of it is similar to the case of simple majority vote. There is however one main difference: under the weighted majority vote 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, associated with bounding the error of getting

$$\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S}) < \varepsilon, \text{ when } W(\hat{S}) > W(S \backslash \hat{S})$$

and

$$\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S}) \geq \varepsilon, \text{ when } W(\hat{S}) = W(S \backslash \hat{S})$$

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

LEMMA 5. *At time t, for set $\hat{S} \subseteq S \subseteq \mathscr{M}$ and its complement $S \backslash \hat{S}$, if $W(\hat{S}) > W(S \backslash \hat{S})$, then at an exploitation step t, $\forall\, 0 < z < 1$,*

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

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

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

The rest of the proof can be found in the appendices.

5. 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, $|\mathscr{W}| < \infty$ 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 of 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 part of the 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 earlier results with the literatures on continuous arms.

6. NUMERICAL EXPERIMENT

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

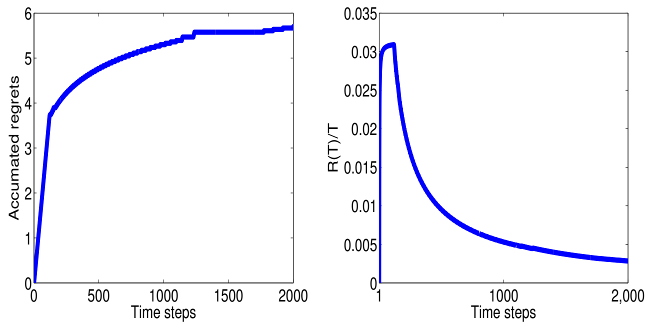
6.1 Simulation study

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 1 (for $a_{\min} = 0.6$) but remain unknown to the labelers. A task arrives at each time t. We assume a unit labeling cost $c = 0.02$. 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:

	L_1	L_2	L_3	L_4	L_5
p_i	0.763	0.781	0.625	0.783	0.727

Table 1: An example of the simulation setup.

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



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

Figure 2: Regret of the LS_OL algorithm.

We first show the accumulative and average regret under the simple majority vote rule in Figure 2. From the set of figures we observe a logarithmic increase of accumulated regret and correspondingly a sub-linear decrease for its average over time. 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 of our 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 Figure 3 in terms of the average reward over time. There is a clear performance improvement after an initialization period (where exploration happens).

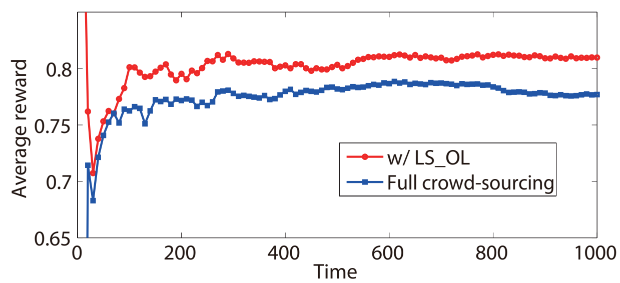


Figure 3: Performance comparison: online labeler selection v.s. full crowd-sourcing (majority vote)

In addition to the logarithmic growth, we are interested in how the performance is affected by the inaccuracy of the crowd expertise. These results are shown in Figure 4. We observe the effect of different choices of $a_{\min} = 0.6, 0.7, 0.8$. 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 error is introduced in the estimation of the labelers' qualities, which results in slower convergence.

We next compare the performance between simple majority vote and weighted majority vote (both with LS_OL). One example trace of accumulated reward comparison is shown in Figure 5; the advantage of weighted majority vote can be seen clearly. We then repeat the set of experiments and average the results over 500 runs; the comparison is shown in Table 2 under different number of candidate labelers (all of their labeling qualities are uniformly generated).

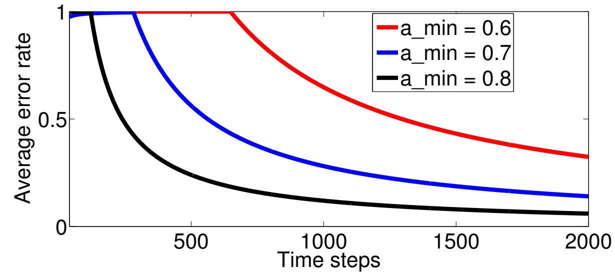


Figure 4: Effect of $a_{\min}$: higher $a_{\min}$ leads to much better performance.

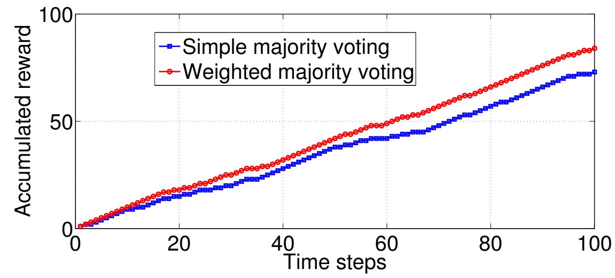


Figure 5: Comparing weighted and simple majority voting within LS_OL.

M	5	10	15	20
Full crowd-sourcing (majority vote)	0.5154	0.5686	0.7000	0.7997
Majority vote w/ LS_OL	0.8320	0.9186	0.9434	0.9820
Weighted majority vote w/ LS_OL	**0.8726**	**0.9393**	**0.9641**	**0.9890**

Table 2: Average reward per labeler: there is a clear gap between with and without using LS_OL.

6.2 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 5 AMTs. The labels are on the scale from 0 to 4 indicating how many scenes are seen from each image, such as field, 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, 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 3, while Figure 6 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 disagreement	348	353	376	338	441

Table 3: Total number of disagreement each AMT has.

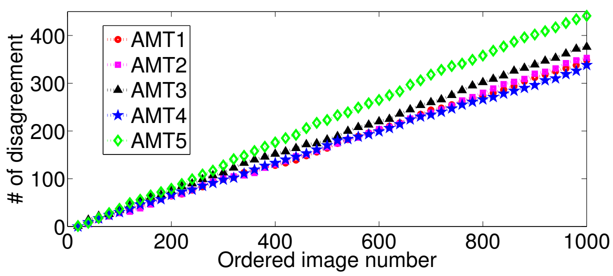


Figure 6: Cumulated number of disagreements.

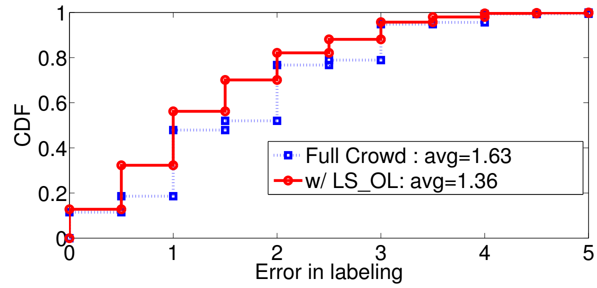


Figure 8: Labeling error distribution comparison.

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 majority vote and with the addition of the detection and filtering procedure discussed in Section 3.5.3, 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 Figure 7 along with a comparison to using the full crowd-sourcing method with majority vote. The algorithm also eventually correctly estimates the best set to consist of AMTs 1, 2, and 4. 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 Figure 8; note the errors are discrete due to the discrete labels. It is also worth noting that under our algorithm the cost is much lower because AMT 5 was soon eliminated, while AMT 3 was only used very infrequently once the correct estimate has been achieved.

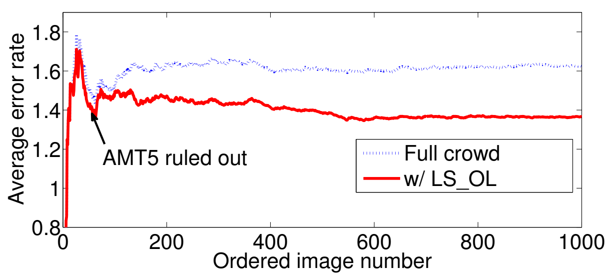


Figure 7: Average error comparison: online labeler selection v.s. full crowd-sourcing.

7. CONCLUSION

To our best knowledge, this is the first work formalizing and addressing the issue of learning labelers' 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 high and low quality labelers over time and select the best set for labeling tasks with $O(\log^2 T)$ regret uniform in time. 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 where labelers' quality varies with contextually different tasks and how to detect and remove malicious labelers when there is a lack of ground-truth. We validate our results via both synthetic and real world AMT data.

Acknowledgment

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

8. 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] Anandkumar, A., Michael, N., Tang, A. K., and Swami, A. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *Selected Areas in Communications, IEEE Journal on 29*, 4 (2011), 731–745.

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

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

[8] 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.

[9] Guille, A., Hacid, H., Favre, C., and Zighed, D. A. Information diffusion in online social networks: A survey. *ACM SIGMOD Record 42*, 2 (2013), 17–28.

[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, H., LIU, K., AND ZHAO, Q. Learning in a changing world: Non-bayesian restless multi-armed bandit. Tech. rep., DTIC Document, 2010.

[18] 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.

[19] NATARAJAN, N., DHILLON, I., RAVIKUMAR, P., AND TEWARI, A. Learning with Noisy Labels. In *Advances in Neural Information Processing Systems* (2013), pp. 1196–1204.

[20] REA, L. M., AND PARKER, R. A. *Designing and conducting survey research: A comprehensive guide*. John Wiley & Sons, 2012.

[21] 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.

[22] 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.

[23] TEKIN, C., AND LIU, M. Online Learning of Rested and Restless Bandits. *Information Theory, IEEE Transactions on 58*, 8 (2012), 5588–5611.

[24] ZHONG, E., FAN, W., AND YANG, Q. Contextual collaborative filtering via hierarchical matrix factorization. In *SDM'12* (2012), pp. 744–755.

[25] ZHU, X., AND GOLDBERG, A. B. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning.

Appendices

Proof of Theorem 1

We prove by contradiction. Suppose m is even and we order all selected users by their labeling capability in descending order $p_1 \geq \ldots \geq p_m$. We now prove

$$\pi(p_1, \ldots, p_{m-1}) \geq \pi(p_1, \ldots, p_m) \tag{16}$$

Consider the following. By adding p_m, the gain for $\pi(p_1, \ldots, p_{m-1})$ is $\frac{p_m \cdot P(T_1)}{2}$, where

$$T_1 = \{\text{\# of correct labels} = \text{\# of wrong labels} - 1\}.$$

That is, only when the number of correct labels is exactly the same as the number of wrong labels less one does adding a m-th correct label change the outcome of the majority vote (in this case there is a tie so the label changes with probability $1/2$); when the former number is smaller or bigger, adding one more vote does not change the results. On the other hand, the loss is $\frac{(1-p_m)\cdot P(T_2)}{2}$, where

$$T_2 = \{\text{\# of correct labels} = \text{\# of wrong labels} + 1\}.$$

We now compare $p_m \cdot P(T_1)$ and $(1-p_m)\cdot P(T_2)$. Within the set T_2, each event is of the form where some labeler i gives the correct label while the rest are half correct and half wrong. Denote this event by ω_i and by ω_{-i} the event that the rest of the labels (given by those other than i) are half right and half wrong. Note for each ω_i there is a corresponding event $\hat{\omega}_i \in T_1$ where i gives the wrong labels while the rest are half correct and half wrong. Since $p_i \geq p_m$ we have

$$(1-p_m)\cdot\, p_i \cdot P(\omega_{-i}) \geq p_m \cdot (1-p_i)\cdot P(\omega_{-i})\,. \tag{17}$$

At the same time, $P(\omega_i) = p_i \cdot P(\omega_{-i}), P(\hat{\omega}_i) = (1-p_i)\cdot P(\omega_{-i})$, i.e., $(1-p_m)\cdot P(\omega_i) \geq p_m \cdot P(\hat{\omega}_i)$. This is true for all ω_i. Therefore we have

$$\begin{aligned}&(1-p_m)\cdot P(T_2)\\ &= (1-p_m)\cdot P(\cup_i\ \omega_i)\\ &= \sum_{\omega_i}(1-p_m)\cdot P(\omega_i)\\ &\geq \sum_{\hat{\omega}_i} p_m \cdot P(\hat{\omega}_i) = p_m \cdot P(T_1)\end{aligned} \tag{18}$$

Therefore we have proved $\pi(p_1, \ldots, p_{m-1}) \geq \pi(p_1, \ldots, p_m)$. Moreover

$$\begin{aligned}&U(\{1,2,\ldots,m-1\}) - U(\{1,2,\ldots,m\})\\ &= \pi(p_1,\ldots,p_{m-1}) - \pi(p_1,\ldots,p_m) > 0.\end{aligned} \tag{19}$$

Therefore a selection of an even number of labelers can always be improved by removing the least accurate labeler, resulting in an odd number of labelers in the selection.

Proof of Theorem 2

Consider a m-set S. Suppose there is a $i \notin S$ and a $j \in S$ such that $p_i > p_j$. Then the probability of making a correct annotation is given by

$$\begin{aligned}&P_S(\text{\# of correct labels} > \text{\# of wrong labels})\\ &= p_j \cdot P(T_1) + (1-p_j)\cdot P(T_2)\end{aligned}$$

where

$$T_1 = \{\text{\# correct label} > \text{\# wrong label} - 1 \text{ in } S\backslash j\} \tag{20}$$

$$T_2 = \{\text{\# correct label} > \text{\# wrong label} + 1 \text{ in } S\backslash j\} \tag{21}$$

Now replace j with i and denote $\hat{S} = S\backslash j \cup \{i\}$ we have

$$\begin{aligned}&P_{\hat{S}}(\text{\# of correct labels} > \text{\# of wrong labels})\\ &= p_i \cdot P(T_1) + (1-p_i)\cdot P(T_2)\,.\end{aligned}$$

It follows that

$$P_{\hat{S}} - P_S = (p_i - p_j)\cdot(P(T_1) - P(T_2))\,. \tag{22}$$

If an event $\omega \in T_2$ we must also have $\omega \in T_1$, thus we have $T_2 \subset T_1$; therefore $P(T_1) - P(T_2) > 0$, and we conclude that $P_{\hat{S}} - P_S > 0$, completing the proof.

Proof of Lemma 1

Denote by $n(T)$ the number of times an exploration phase has been activated up to time T. Since for labeler i there is at most $D_1(T)\cdot D_2(T)$ number of exploration phases, we have

$$\begin{aligned} n(T) &= \sum_{i=1}^{T} I_{\text{at least one task in exploration phase at } t} \\ &\leq \sum_{k=1}^{D_1(t)} \sum_{i=1}^{T} I_{\text{task } k \text{ in reassignment phase at } t} \leq D_1(T)\cdot D_2(T) \ , \end{aligned}$$

where the first inequality comes from union bound. Then

$$E[R_e(T)] \leq U(S^*)\cdot n(T) = U(S^*)\cdot (D_1(T)\cdot D_2(T)) \ .$$

Proof of Lemma 3

Firstly notice via union bound we have the following bound at any time t:

$$E[\mathscr{E}_1(t)] \leq \sum_{\substack{m=1 \\ m \text{ odd}}}^{M} P(\tilde{U}(S^m) \geq \tilde{U}(S^*)) \ . \tag{23}$$

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 6. *The probability of using a sub-optimal selection S^m is bounded as follows,*

$$\begin{aligned} P(\tilde{U}(S^m) \geq \tilde{U}(S^*)) \leq\ & P(\tilde{U}(S^m) > U(S^m) + \varepsilon) \\ &+ P(\tilde{U}(S^*) < U(S^*) - \varepsilon) \ , \end{aligned} \tag{24}$$

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

$$\begin{aligned} & P(|\tilde{U}(S) - U(S)| > \varepsilon) \\ & \leq n(S)\cdot \sum_{i\in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|}) \ . \end{aligned} \tag{25}$$

We shall now use the above lemma; its own proof is given later in this appendix.

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

$$\begin{aligned} & P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|}) \\ &= \underbrace{P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|} \mid \frac{\sum_{k:k\in E(t)} I_{y_k^*=0}}{|E(t)|} \leq \frac{\alpha\cdot\varepsilon}{t^z})}_{\textbf{Term 1}} \\ &\quad \cdot P(\frac{\sum_{k:k\in E(t)} I_{y_k^*=0}}{|E(t)|} \leq \frac{\alpha\cdot\varepsilon}{t^z}) \\ &+ P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|} \mid \frac{\sum_{k:k\in E(t)} I_{y_k^*=0}}{|E(t)|} > \frac{\alpha\cdot\varepsilon}{t^z}) \\ &\quad \cdot \underbrace{P(\frac{\sum_{k:k\in E(t)} I_{y_k^*=0}}{|E(t)|} > \frac{\alpha\cdot\varepsilon}{t^z})}_{\textbf{Term 2}} \end{aligned} \tag{26}$$

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 follows:

$$\begin{aligned} & P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S)\cdot|S|} \mid \frac{\sum_{k:k\in E(t)} I_{y_k^*=0}}{|E(t)|} \leq \frac{\alpha\cdot\varepsilon}{t^z}) \\ & \leq P(|\tilde{p}_i - p_i| > (\frac{1}{n(S)\cdot|S|} - \frac{\alpha}{t^z})\cdot\varepsilon) \\ & \leq 2\cdot e^{-2((\frac{1}{n(S)\cdot|S|} - \frac{\alpha}{t^z})\cdot\varepsilon)^2\cdot D_1(t)} \leq \frac{2}{t^2} \ , \end{aligned} \tag{27}$$

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

$$\begin{aligned} P(\frac{\sum_{k:k\in E(t)} I_{y_k^*=0}}{|E(t)|} > \frac{\alpha\cdot\varepsilon}{t^z}) &\leq \frac{\frac{E[\sum_{k:k\in E(t)} I_{y_k^*=0}]}{|E(t)|}}{\frac{\alpha\cdot\varepsilon}{t^z}} \\ &= \frac{\frac{\sum_{k:k\in E(t)} E[I_{y_k^*=0}]}{|E(t)|}}{\frac{\alpha\cdot\varepsilon}{t^z}} \ , \end{aligned} \tag{28}$$

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

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

where $\hat{N}_k(t)$ is the number of feedbacks received for task k up-to 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

$$P(|\tilde{U}(S) - U(S)| > \varepsilon) \leq n(S)\cdot|S|\cdot(\frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}}) \ . \tag{29}$$

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

Proof of Lemma 4

We have the following fact:

$$\begin{aligned} E[\mathscr{E}_2(t)] &\leq P(\cup_{i\in\mathscr{M}} |\tilde{p}_i - p_i| > \varepsilon) \\ &\leq \sum_{i\in\mathscr{M}} P(|\tilde{p}_i - p_i| > \varepsilon) \ . \end{aligned}$$

This is because if $|\tilde{p}_i - p_i| \leq \varepsilon, \forall i$ then we must have for $p_i > p_j$,

$$\tilde{p}_i - \tilde{p}_j \geq p_i - \varepsilon - p_j - \varepsilon > 0,$$

which means there is no error in ordering. Similarly as above we have

$$P(|\tilde{p}_i - p_i| > \varepsilon) \leq \frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}} \ . \tag{30}$$

Therefore,

$$E[\mathscr{E}_2(t)] \leq M(\frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}}) \ . \tag{31}$$

Proof of Lemma 6

We first bound the inequality in Eqn.(24). To see why this inequality is true, consider the following fact

$$\{\omega : \tilde{U}(S^m) \geq \tilde{U}(S^*)\} \subseteq$$
$$\{\omega : \tilde{U}(S^m) > U(S^m) + \varepsilon\} \cup \{\omega : \tilde{U}(S^*) < U(S^*) - \varepsilon\} \;, \quad (32)$$

since if $\tilde{U}(S^m) \leq U(S^m) + \varepsilon,\; \tilde{U}(S^*) \geq U(S^*) - \varepsilon$, we then have

$$\tilde{U}(S^m) - \tilde{U}(S^*) \leq U(S^m) + \varepsilon - U(S^*) + \varepsilon$$
$$\leq -\Delta_{\min} + 2\varepsilon < -\Delta_{\min} + \Delta_{\min} = 0 \;,$$

which contradicts the fact that $\tilde{U}(S^m) \geq \tilde{U}(S^*)$. Thus

$$P(\tilde{U}(S^m) \geq \tilde{U}(S^*)) \leq P(\tilde{U}(S^m) > U(S^m) + \varepsilon)$$
$$+ P(\tilde{U}(S^*) < U(S^*) + \varepsilon) \;,$$

by the union bound. The bounding effort then reduces to bounding each of above probabilities. Note that for any set S, plug in $U(S)$ we have

$$|\tilde{U}(S) - U(S)| = |\sum_{S': S' \subseteq S,\; |S'| \geq \lceil \frac{|S|}{2} \rceil} (\prod_{i \in S'} \tilde{p}_i \prod_{S \backslash S'} (1 - \tilde{p}_j))$$
$$- \prod_{i \in S'} p_i \prod_{S \backslash S'} (1 - p_j))| \;. \quad (33)$$

Therefore

$$P(|\tilde{U}(S) - U(S)| > \varepsilon) = P(\sum_{S': S' \subseteq S,\; |S'| \geq \lceil \frac{|S|}{2} \rceil} |\prod_{i \in S'} \tilde{p}_i \prod_{S \backslash S'} (1 - \tilde{p}_j)$$
$$- \prod_{i \in S'} p_i \prod_{S \backslash S'} (1 - p_j)| > \varepsilon)$$
$$\leq \sum_{S': S' \subseteq S,\; |S'| \geq \lceil \frac{|S|}{2} \rceil} P(|\prod_{i \in S'} \tilde{p}_i \prod_{S \backslash S'} (1 - \tilde{p}_j)$$
$$- \prod_{i \in S'} p_i \prod_{S \backslash S'} (1 - p_j)| > \frac{\varepsilon}{n(S)}) \;,$$

where the last inequality comes from the union bound. We further use the following results (which can be proved separately but the proof is omitted) to separate the above product terms into summations.

LEMMA 7. *For $k \geq 1$ and two sequences $\{l_i\}_{i=1}^m$ and $\{q_i\}_{i=1}^m$ and $0 \leq l_i, q_i \leq 1, \forall i = 1, ..., k.$, we have*

$$|\prod_{i=1}^m l_i - \prod_{j=1}^m q_j| \leq \sum_{i=1}^m |l_i - q_i| \;. \quad (34)$$

Using this result, we have

$$|\prod_{i \in S'} \tilde{p}_i \prod_{j \in S \backslash S'} (1 - \tilde{p}_j) - \prod_{i \in S'} p_i \prod_{j \in S \backslash S'} (1 - p_j)|$$
$$\leq \sum_{i \in S'} |\tilde{p}_i - p_i| + \sum_{j \in S \backslash S'} |(1 - \tilde{p}_j) - (1 - p_j)|$$
$$= \sum_{i \in S} |\tilde{p}_i - p_i| \;.$$

Therefore using the union bound we have

$$P(|\prod_{i \in S'} \tilde{p}_i \prod_{S \backslash S'} (1 - \tilde{p}_j) - \prod_{i \in S'} p_i \prod_{S \backslash S'} (1 - p_j)| > \frac{\varepsilon}{n(S)})$$
$$\leq \sum_{i \in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|}) \;.$$

Therefore summing up all of the above we have

$$P(|\tilde{U}(S) - U(S)| > \varepsilon)$$
$$\leq \sum_{S': S' \subseteq S,\; |S'| \geq \lceil \frac{|S|}{2} \rceil} \sum_{i \in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|})$$
$$= n(S) \cdot \sum_{i \in S} P(|\tilde{p}_i - p_i| > \frac{\varepsilon}{n(S) \cdot |S|}) \;. \quad (35)$$

Proof of Theorem 4

We prove this by contradiction. Suppose there exists a pair (i, j), $i \in S, j \notin S$ such that $p_i < p_j$. We discuss the following cases. First of all as we already noted we have $\log \frac{p_i}{1-p_i} < \log \frac{p_j}{1-p_j}$. Consider the following fact the probability for correct labeling is given by

$$P_c = p_i \cdot P(T_1) + (1 - p_i) \cdot P(T_2) + \frac{P(T_3)}{2} \;, \quad (36)$$

where

$$T_1 = \{\sum_{u \in S_c} \log \frac{p_u}{1 - p_u} > \sum_{e \in S_w} \log \frac{p_e}{1 - p_e} - \log \frac{p_i}{1 - p_i}\} \;,$$
$$T_2 = \{\sum_{u \in S_c} \log \frac{p_u}{1 - p_u} > \sum_{e \in S_w} \log \frac{p_e}{1 - p_e} + \log \frac{p_i}{1 - p_i}\} \;,$$

and

$$T_3 = \{\text{A tie occurs}\}, \quad (37)$$

where $S_c \neq S_w$ and $S_c \cup S_w = \mathscr{M} - \{i\}$, indicating the set of correct and wrong labelers respectively. Essentially the first two events correspond to cases when there is a majority group (including and excluding i respectively) and T_3 corresponds to the case when there is a tie.

Changing p_i to p_j since

$$P(T_1^j) \geq P(T_1),\; P(T_2) \geq P(T_2^j) \;,$$

if $p_i > 0.5$, where $T_q^j, q \in \{1, 2\}$ correspond to $T_q, q \in \{1, 2\}$ by replacing i with j. And we have

$$p_j \cdot P(T_1^j) + (1 - p_j) \cdot P(T_2^j)$$
$$- p_i \cdot P(T_1) - (1 - p_i) \cdot P(T_2)$$
$$\geq (p_j - p_i) \cdot (P(T_1) - P(T_2)) \geq 0 \;.$$

For T_3 consider the case p_i is in S_c. Then changing p_i to p_j will break the equilibrium and the probability of a correct output will become

$$p_j \cdot P(S_c) \cdot P(S_w) > p_i \cdot P(S_c) \cdot P(S_w) = \frac{P(T_3)}{2} \;, \quad (38)$$

where $P(S_c), P(S_w)$ correspond to the probabilities from the correct and wrong labelers, i.e.,

$$P(S_c) = \sum_{u \in S_c} p_u,\; P(S_w) = \sum_{e \in S_w} (1 - p_e) \;, \quad (39)$$

and the last inequality comes from the fact that in the equal case the probabilities of the label being either 0 or 1 are equivalent with each other.

Proof of Lemma 5

First of all we have

$$P(\tilde{W}(\hat{S}) - \tilde{W}(S \backslash \hat{S}) < \varepsilon) \leq P(\tilde{W}(\hat{S}) - W(\hat{S}) < -\varepsilon/2)$$
$$+ P(\tilde{W}(S \backslash \hat{S}) - W(S \backslash \hat{S}) > \varepsilon/2) \;.$$

This is because otherwise if

$$\tilde{W}(\hat{S}) - W(\hat{S}) \geq -\varepsilon/2,$$
$$\tilde{W}(S\backslash\hat{S}) - W(S\backslash\hat{S}) < \varepsilon/2\ ,$$

we have

$$\tilde{W}(\hat{S}) - \tilde{W}(S\backslash\hat{S}) \geq W(\hat{S}) - W(S\backslash\hat{S}) - \varepsilon \geq \varepsilon\ , \tag{40}$$

which gives us a contradiction.

Consider each term above we have,

$$\begin{aligned}
&P(\tilde{W}(\hat{S}) - W(\hat{S}) < -\varepsilon/2) \\
&\leq \sum_{i\in\hat{S}} P(|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| > \frac{\varepsilon}{2|\hat{S}|}) \\
&\leq \sum_{i\in\hat{S}} P(|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| > \frac{\varepsilon}{2|\hat{S}|}\,||\tilde{p}_i - p_i| \geq \frac{\varepsilon}{4C|\hat{S}|}) \\
&\qquad \cdot P(|\tilde{p}_i - p_i| \geq \frac{\varepsilon}{4C|\hat{S}|}) + \\
&\sum_{i\in\hat{S}} P(|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| \geq \frac{\varepsilon}{2|\hat{S}|}\,||\tilde{p}_i - p_i| < \frac{\varepsilon}{4C|\hat{S}|}) \\
&\qquad \cdot P(|\tilde{p}_i - p_i| < \frac{\varepsilon}{4C|\hat{S}|}) \\
&\leq \sum_{i\in S} P(|\tilde{p}_i - p_i| \geq \frac{\varepsilon}{4C|\hat{S}|}) \leq |\hat{S}|(\frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}})\ .
\end{aligned}$$

Since

$$D_1(t) \geq 1/(\frac{1}{\max_m 4C\cdot m} - \alpha)\log t,\ \alpha < \frac{1}{\max_m 4C\cdot m}\ ,$$

as well as the fact that when $|\tilde{p}_i - p_i| \leq \frac{\varepsilon}{4C|\hat{S}|}$ and

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

we have

$$|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| \leq 2C\cdot|\tilde{p}_i - p_i| < \frac{\varepsilon}{2|\hat{S}|}\ ,$$

where we have used the following results.

LEMMA 8. *With $\tilde{p}_i, p_i$ bounded away from 0 and 1 we have,*

$$|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| \leq 2C\cdot|\tilde{p}_i - p_i|\ , \tag{41}$$

where C is a constant satisfying,

$$C > \max_i \max\{\frac{1}{p_i}, \frac{1}{\tilde{p}_i}, \frac{1}{1-p_i}, \frac{1}{1-\tilde{p}_i}\}. \tag{42}$$

Similarly we have

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

Combine above we have

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

For the other case when $W(\hat{S}) = W(S\backslash\hat{S})$ we have,

$$\begin{aligned}
&P(|\tilde{W}(\hat{S}) - \tilde{W}(S\backslash\hat{S})| > \varepsilon) \leq P(|\tilde{W}(\hat{S}) - W(\hat{S})| \geq \varepsilon/2) \\
&\qquad + P(|\tilde{W}(S\backslash\hat{S}) - W(S\backslash\hat{S})| \geq \varepsilon/2) \\
&\leq |S|\cdot(\frac{2}{t^2} + \frac{1}{\alpha\cdot\varepsilon\cdot t^{2-z}})\ ,
\end{aligned} \tag{43}$$

where the second inequality is established similarly as in the first case.

Proof of Lemma 7

We prove the claim by induction. Notice when $m = 1$ the inequality holds trivially. When $m = 2$ we have

$$\begin{aligned}
&|l_1\cdot l_2 - q_1\cdot q_2| \\
&= |(l_1 - q_1)\cdot\frac{l_2+q_2}{2} + (l_2 - q_2)\cdot\frac{l_1+q_1}{2}| \\
&\leq |(l_1 - q_1)\cdot\frac{l_2+q_2}{2}| + |(l_2 - q_2)\cdot\frac{l_1+q_1}{2}| \\
&= |l_1 - q_1|\cdot|\frac{l_2+q_2}{2}| + |l_2 - q_2|\cdot|\frac{l_1+q_1}{2}| \\
&\leq |l_1 - q_1| + |l_2 - q_2|\ .
\end{aligned} \tag{44}$$

The last inequality used the fact

$$|\frac{l_1+q_1}{2}| \leq 1,\quad |\frac{l_2+q_2}{2}| \leq 1\ .$$

Suppose the inequality holds for m.

$$\begin{aligned}
&|\prod_{i=1}^{m+1} l_i - \prod_{j=1}^{m+1} q_j| = |\prod_{i=1}^{m} l_i\cdot l_{m+1} - \prod_{j=1}^{m} q_j\cdot q_{m+1}| \\
&\leq |\prod_{i=1}^{m} l_i - \prod_{j=1}^{m} q_j| + |l_{m+1} - q_{m+1}| \\
&\leq \sum_{i=1}^{m+1} |l_i - q_i|\ ,
\end{aligned} \tag{45}$$

where the second inequality comes from the induction basis for $m = 2$, since

$$0 \leq \prod_{i=1}^{m} l_i, \prod_{j=1}^{m} q_j \leq 1\ ,$$

and the last inequality uses the induction hypothesis.

Proof of Lemma 8

Observe the following facts:

$$\begin{aligned}
&|\log\frac{\tilde{p}_i}{1-\tilde{p}_i} - \log\frac{p_i}{1-p_i}| \\
&= |\log\tilde{p}_i - \log p_i + \log(1-p_i) - \log(1-\tilde{p}_i)| \\
&\leq |\log\tilde{p}_i - \log p_i| + |\log(1-p_i) - \log(1-\tilde{p}_i)| \\
&\leq 2C|\tilde{p}_i - p_i|\ ,
\end{aligned} \tag{46}$$

since all four terms are bounded from 0 and the last inequality comes from classical inequality of $\log(\cdot)$ functions.