Active Crowdsourcing for Annotation

Shuji Hao
Joint NTU-UBC Research Center (LILY),IGS
Nanyang Technological University, Singapore
Email: haos0001@e.ntu.edu.sg

Chunyan Miao
School of Computer Engineering
Nanyang Technological University, Singapore
Email: ascymiao@ntu.edu.sg

Steven C.H. Hoi
School of Information Systems
Singapore Management University, Singapore
Email: chhoi@smu.edu.sg

Peilin Zhao
Institute for Infocomm Research
A*STAR, Singapore
Email: zhaop@i2r.a-star.edu.sg

***Abstract*—Crowdsourcing has shown great potential in obtaining large-scale and cheap labels for different tasks. However, obtaining reliable labels is challenging due to several reasons, such as noisy annotators, limited budget and so on. The state-of-the-art approaches, either suffer in some noisy scenarios, or rely on unlimited resources to acquire reliable labels. In this article, we adopt the learning with expert (AKA worker in crowdsourcing) advice framework to robustly infer accurate labels by considering the reliability of each worker. However, in order to accurately predict the reliability of each worker, traditional learning with expert advice will consult with external oracles (AKA domain experts) on the true label of each instance. To reduce the cost of consultation, we proposed two active learning approaches, margin-based and weighted difference of advices based. Meanwhile, to address the problem of limited annotation budget, we proposed a reliability-based assigning approach which actively decides who to annotate the next instance based on each worker's cumulative performance. The experimental results both on real and simulated datasets show that our algorithms can achieve robust and promising performance both in the normal and noisy scenarios with limited budget.**

I. Introduction

For most of the supervised machine learning algorithms, a large-scale of training data with golden labels is usually required. However, labeling the data is usually costly and time-consuming. To speed up the labeling process and obtaining reliable labels, crowdsourcing platforms, such as Amazon Mechanical Turk [1], CrowdFlower [2] and Baidu Test [3] and so on, are developed to easily obtain large-scale and reliable labels. However, obtaining reliable labels from crowdsourcing faces two main challenges. First, the annotations are usually noisy due to different knowledge bases of workers, also known as annotators, varied difficulty of each task and so on [24]. This gives rise to the question how to combine the noisy annotations to get finalized accurate labels. Second, the budget size is usually limited, and obtaining unlimited annotations for each task is costly and impractical.

To tackle the problem of noisy annotations, several approaches in literature are proposed. One promising approach is repeated labeling [22]. The basic idea is to score the workers by their agreement with other workers. Most of the existing work followed this approach by treating the true label and workers' reliability as latent variables, and inferred these variables by building joint statistical models [25], [26], [27]. These algorithms are attractive because of the promising performances they shown in real crowdsourcing datasets without referring gold standard questions. However, these algorithms are heavily relying on the assumption that the majority of workers are correct, which may not be always true [18]. These unsupervised algorithms would perform as bad as the majority voting (MV) algorithm when the assumption could not be met, such as in group-cheating scenarios, where group of people cheat together by giving the same and noisy label for each instance. To tackle the budget issue, Welinder et al. [25] proposed to actively decide who to label by building a reliable worker list and a noisy worker list. However, this algorithm also suffers the same issue of requiring most of the workers to be reliable.

Considering the group-cheating and limited budget issues, we argue that it is necessary to build some gold standard questions with known answers, based on which workers' reliability can be accurately estimated and thus workers' annotations can be properly aggregated to get finalized accurate labels even in extreme noisy scenarios. Meanwhile, the usage of gold standard questions should be limited considering their high prices and the limited budget. Another argument we made here is that it is unsuitable to uniformly allocate tasks to each worker once we obtain their reliabilities.

In this article, we adopt the learning with expert (corresponding to worker in crowdsourcing) advice framework to tackle the issues of noisy workers and limited budget on crowdsourcing tasks. However, traditional learning with expert advice will consult with external oracles on the true labels of each instance. To reduce the cost of consultation, we proposed two active learning approaches, margin-based and weighted difference of advices based. To further address the limited budget issue, we proposed a reliability-based allocation approach which actively decides who to annotate the given instance based on each worker's cumulative performance. The experimental results both on real and simulated datasets show that our algorithms can achieve robust and promising performance both in the normal and extreme noisy scenarios with limited budget.

[1] https://www.mturk.com/mturk/
[2] http://www.crowdflower.com/
[3] http://test.baidu.com/crowdtest/

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents the proposed framework and algorithms. Section 4 discusses our empirical study and Section 5 concludes this work.

II. Related Work

In this section, we review the related work on inferring ground label for annotation tasks and the active learning approaches related to our proposed algorithms.

Crowdsourcing, as an approach to collect large-scale and cheap labeled data, has attracted great attention in recent years [2], [9], [14], [16], [23], [28]. One of the critical problems is how to properly aggregate the noisy annotations to get accurate finalized labels. The most common method to infer the ground label is majority voting [22]. However, this approach would suffer when the number of noisy annotators is larger than the number of reliable annotators, which is common in the crowdsourcing platforms. The idea of majority voting approach is taken one step further by looking at the consistency between annotators. Most of recent work follows this line by modeling the true label, worker's ability, and the task's difficulty as the latent variables within graph models, by which the problem is transferred to inferring latent variables with EM-style algorithms. Dawid and Skene [8] modeled the reliabilities of workers using a confusing matrix. Whitehill et.al. [26] further modeled both worker reliability and image difficulty. However, the algorithm would be worse than the majority voting when the variance of reliabilities between workers is high [25]. To tackle this issue, Welinder et al. [25] further introduced a high-dimensional concept of image difficulty and annotator bias. These algorithms could achieve promising performance without explicitly evaluating workers' reliability by querying golden label of sample instances. However, these algorithms would suffer in noisy scenarios, such as group of people cheating together to give the same labels [10].

Active learning has been extensively studied both in supervised and unsupervised learning algorithms [1], [5], [20]. Interested users are recommended to refer the survey article written by Settles [21]. Here we mainly focus on the literature which is close to our active learning strategies and the work which apply active learning approaches to the annotation tasks. Due to its robustness and effectiveness, margin-based approach has been widely used in literature [15]. Cesa-Bianchi et al. [4] further applied the margin-based approach in the online setting. Along this direction, we proposed our first active learning algorithm, where the weighted annotation is defined as the margin. Related to our second active learning approach, Zhao et al. [30] proposed active learning with expert advice by considering the difference of advices from different experts, however, the proposed strategies assume the advice is a float number in $[0, 1]$, which is infeasible in the crowdsourcing platforms, where the annotation is usually a categorical value, such as $\{0, 1\}$ in the binary case. Along this direction, we propose a strategy based on the weighted difference of advices by considering the cumulative performance of each expert. Several researchers also studied the problem of applying active learning approaches on crowdsourcing to alleviate the cost of labeling [3], [29]. However, these active learning algorithms in literature are focusing on training specific machine learning models, while our proposed algorithms are focusing on inferring the true labels for any machine learning models.

III. A Learning Framework for Active Crowdsourcing

A. Overview

Consider a real world online binary annotation task, such as image annotation (the workers are asked to label whether an image contains a bird or not), where images (AKA instances) arrive sequentially. Our goal is to construct a predictive model which can accurately infer the true label of the image given several noisy labels annotated by a pool of noisy workers. In general, this can be formulated as the framework of learning with expert advice for binary classification, where each expert corresponds to a noisy worker, and a piece of advice corresponds to a label given by a noisy worker. Based on the labels given by the workers, the goal of learning with expert advice is to train a forecaster which can correctly combine the noisy annotations to predict a correct label of an image. In particular, on each learning round, all the workers first receive a new coming image, and then give their annotations, based on which the forecaster predicts the label of the image. After that, forecaster receives the ground-truth of the image from external oracles (who are usually domain experts), and both the workers and forecaster suffer some positive loss based on the ground-truth and their predicted labels. At last, forecaster updates the reliability of each worker in order to make better prediction in future. It is natural to apply the learning with expert advice forecasters, such as "Exponentially Weighted Average Forecaster" (EW) [5] and "Greedy Forecaster" (GF) [5], to the annotation tasks in the crowdsourcing platforms. However, it is impractical to directly apply the EW and GF forecasters to the problem due to several reasons. First, for the regular forecasters, each instance in the annotation task would need all workers to annotate. However, for a real annotation task, the budget is usually limited and noisy workers commonly exist, which make it costly and unnecessary to obtain all workers' annotations for each instance. Second, for the traditional forecasters, they require that all instances at last receive the ground-truth from the external oracles, such as domain experts. However, obtaining the ground-truth from the domain experts is much more costly than obtaining the annotations from noisy

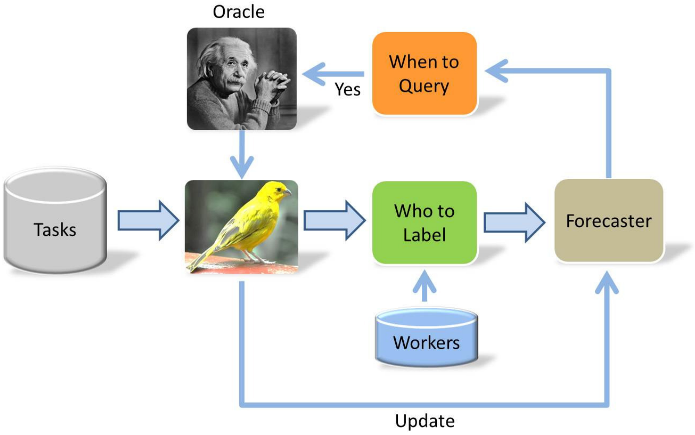


Fig. 1. A Framework of Budget Active Learning with Expert Advice (BALEA)

workers, so it is impractical to obtain a large set of standard questions with ground-truth in order to accurately predict the reliability of each worker. To address these issues, we propose the Budget Active Learning with Expert Advice (BALEA) framework, as shown in Figure 1.

In general, the proposed BALEA framework attempts to conquer the following challenges with a systematic and synergic way: (i) in the "Who to Label" module, the challenge is how to actively decide which workers to label the incoming instance in order to reduce cost and get reliable annotations; and (ii) in the "When to Query" module, the challenge is how to actively decide whether to query domain experts in order to update the workers' reliabilities. To conquer the first challenge, our basic idea is considering the cumulative performance of each worker when allocating the incoming instances. For the second challenge, we propose two active learning strategies which can greatly reducing the cost in query domain experts. Before presenting our detailed algorithms, we first give a formal formulation of the problem and introduce the regular EW and GF forecasters.

B. Problem Formulation

We first define the problem of learning the reliabilities (AKA competences or weights) of workers. For simplicity, we consider the binary annotation task in this article.

For a fixed pool of N annotators, the goal is to train an online learner (AKA forecaster, which describes the way of how to combine the annotations) from a sequence of instances $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ where $\mathbf{x}_t$ is an instance in the annotation task. After receiving $\mathbf{x}_t$, N annotators give their annotations $\{f_t^i : \mathbb{R}^d \to \{0,1\} | i = 1, \ldots, N\}$, based on which the learner makes the final prediction $p_t \in [0,1]$, p_t can be treated as the weighted majority voting or margin of forecaster on $\mathbf{x}_t$. After making the prediction, the learner is revealed with the ground-truth $y_t \in \{0,1\}$ from an external oracle. At last, we evaluate the performance of each annotator according to some non-negative loss functions between each worker's label and the ground-truth.

We formulate the problem above as the problem of learning with expert advice. The label given by each worker corresponds to the advice of each expert. We adopt the absolute loss function to score each worker's performance. For example, the i-th worker's loss on $\mathbf{x}_t$ is defined as $\ell(f_t^i, y_t) = |f_t^i - y_t|$, and the forecaster's loss is defined as $\ell(p_t, y_t) = |p_t - y_t|$. The cumulative losses suffered by the i-th expert and the forecaster are computed respectively as follows:

$$L_T^i = \sum_{t=1}^{T} \ell(f_t^i, y_t), \quad L_T = \sum_{t=1}^{T} \ell\left(p_t, y_t\right).$$

The regret of i-th expert is defined as the loss difference between the i-th expert and the forecaster:

$$R_T^i = L_T^i - L_T, i = 1, \ldots, N. \tag{1}$$

C. EW and GF Forecasters

To aggregate the advices of experts, a natural strategy of combining advices is weighted average prediction strategy. More specifically, the forecaster makes the prediction of $\mathbf{x}_t$ as follows:

$$p_t = \frac{\sum_{i=1}^N w_{t-1}^i f_t^i}{\sum_{i=1}^N w_{t-1}^i}, \tag{2}$$

where w_{t-1}^i is the weight computed at time $t-1$ for the i-th expert. The intuitive idea of learning the weight is to assign large weights for those experts with low loss and small weights for those with high loss.

Next we introduce a special case that leads to the well-known forecaster, known as "Exponentially Weighted average forecaster" (EW) [5]. In particular, by choosing

$$w_{t-1}^i = \frac{\exp(-\eta L_{t-1}^i)}{\sum_{j=1}^N \exp(-\eta L_{t-1}^j)},$$

where $\eta > 0$ is a parameter to control the learning rate, the EW forecaster makes the following prediction:

$$p_t = \sum_{i=1}^{N} w_{t-1}^i f_t^i. \tag{3}$$

In addition to the weighted average forecaster, we also consider another kind of forecaster, known as the "Greedy Forecaster" (GF) [5], which makes the following prediction:

$$p_t = \pi_{[0,1]} \left(\frac{1}{2} + \frac{1}{2\eta} \ln \frac{E(i,1)}{E(i,0)} \right), \tag{4}$$

where $\pi_{[0,1]}(\cdot) = \max(0, \min(1, \cdot))$, $E(i,1) = \sum_{i=1}^N \exp(-\eta L_{t-1}^i - \eta \ell(f_t^i, 1))$ and $E(i,0) = \sum_{i=1}^N \exp(-\eta L_{t-1}^i - \eta \ell(f_t^i, 0))$.

According to the existing studies [5], we have the following theorem for the regret bounds of the above EW and GF algorithms.

Theorem 1: Consider the loss function $\ell(p, y) = |p - y|$, then for any $T > 0$ and $\eta > 0$, and for all $y_1, \ldots, y_T \in \{0,1\}$, the regrets of both the EW and GF algorithms are bounded from above as follows:

$$R_T = L_T - \min_{1 \le i \le N} L_T^i \le \frac{\ln(N)}{\eta} + \frac{\eta T}{8}.$$

By choosing $\eta = \sqrt{8 \ln N / T}$, the regret is bounded from above by $\sqrt{(T/2) \ln N}$.

The above theorem shows both the EW and GF algorithms satisfy the Hannan consistency [12], i.e., $R_T \le o(T)$, which guarantees that the learner could trace the best annotator as T grows.

D. Who to Label

For the "Who to Label" module in Figure 1, the basic idea is to give more chance to the reliable workers. Here, the metric to evaluate the reliability of each worker is the cumulative loss suffered. The higher the loss L_t^i is, the less reliable of the i-th worker is.

Specifically, we define the **unreliability** of i-th worker on t-th round as follows:

$$u_t^i = \frac{\exp(L_{t-1}^i)}{\sum_{i=1}^N \exp(L_{t-1}^i)}. \tag{5}$$

We can see $u_t^i \in (0,1)$ and $\sum_i^N u_t^i = 1$. The larger u_t^i is, the more **unreliable** the i-th worker is.

For the incoming $\mathbf{x}_t$, we draw a Bernoulli random variable $S_t^i \in \{0,1\}$ for i-th worker with probability

$$a_t^i = \frac{\sigma}{\sigma + u_t^i}, \tag{6}$$

where $\sigma > 0$ is a parameter to trade off the budget size and the number of annotations the t-th instance could get. If $S_t^i = 1$, the i-th worker would get the chance to annotate $\mathbf{x}_t$. If $S_t^i = 0$, the algorithm will skip the i-th worker. From the equation, we can see, the more **unreliable** the i-th work is, the more chance the work would **not** get the t-th annotation task.

E. When to Query

For the traditional learning with expert advice, the ground-truth is always obtained from the domain experts for each instance. However, consulting with domain experts is highly costly, so the key challenge is how to design effective *Active learner* for the "When to Query" module in Figure 1. By observing the Equation (3) and (4), we could treat the p_t as the confidence of forecaster on instance $\mathbf{x}_t$, therefore, we adopt a simple yet effective active learning scheme to decide when $\mathbf{x}_t$ should be queried.

Specifically, for both the EW and GF forecasters, at the t-th round, the algorithm decides when $\mathbf{x}_t$ should be queried according to a Bernoulli random variable $Z_t \in \{0,1\}$ with probability

$$q_t = \frac{\delta}{\delta + |p_t|}, \tag{7}$$

where p_t is computed based on Equation (3) or (4), and $\delta > 0$ is a sampling parameter to control the query ratio to trade off the budget size and the number of consultation with domain experts. This strategy is similar to the margin-based active learning [21], and has been used in online classification problem [4]. If $Z_t = 1$, the true label y_t provided by domain experts is revealed to the forecaster, and the algorithm will update workers' reliabilities. If $Z_t = 0$, the true label will not be queried and the reliabilities of workers will not be updated.

We also propose another active learning strategy by exploring the weighted consistence of workers' annotations. In the work [30], the authors assumed the advice given by each expert is a float point in the range of $[0,1]$. However, for the annotation tasks in the crowdsourcing platforms, the label given by each worker is in $\{0,1\}$, which makes the proposed strategies infeasible. Besides, the proposed strategies [30] are only based on the annotations f_t^i on current instance $\mathbf{x}_t$, and the cumulative performance of each worker shown in Equation (5) is not considered, which makes the strategies suffer when noisy workers exist. In this article, we proposed two active learning strategies by considering both workers' annotations and their cumulative performances. For an instance $\mathbf{x}_t$, we denote $\hat{p}_t$ as the prediction of forecaster which is trained based on queried instances until t-th round, and p_t as the prediction of forecaster which is trained based on all instance until t-th round. The basic idea is to design querying strategies which guarantee a small difference between $\hat{p}_t$ and p_t. Formally, for the EW forecaster, if the following theorem is satisfied, the ground-truth of $\mathbf{x}_t$ is **unnecessary** to be revealed ($Z_t = 0$), vice versa.

Theorem 2: For a small constant $\beta > 0$, if the following condition is satisfied, then $|p_t - \widehat{p}_t| \le \beta$

$$\left| \frac{\sum_{i=1}^N \sum_{j=1}^N \gamma_{t-1}^{i,j}(f_t^i - f_t^j)}{\sum_{i=1}^N \sum_{j=1}^N \gamma_{t-1}^{i,j}} \right| \le \beta, \tag{8}$$

where $\gamma_{t-1}^{i,j} = \exp\left(-\eta\left[\widehat{L}_{t-1}^i + \widehat{H}_{t-1}^i + \widehat{L}_{t-1}^j\right]\right)$, $\widehat{L}_{t-1}^i$ and $\widehat{H}_{t-1}^i$ are the losses suffered of i-th worker until t-th round on queried and un-queried instances, respectively. $\beta > 0$ is a small constant to control the query ratio.

Similarly, for the GF forecaster, if the following theorem is satisfied, the ground-truth of $\mathbf{x}_t$ is **unnecessary** to be revealed ($Z_t = 0$), vice versa.

Theorem 3: For a small constant $\beta > 0$, if the following condition is satisfied, then $|p_t - \widehat{p}_t| \le \beta$.

$$\frac{1}{2\eta} \ln \frac{\sum_{i=1}^N \mu_t^i \exp\left(2\eta(f_t^i - p_t)\right)}{\sum_{i=1}^N \mu_t^i} \le \beta, \tag{9}$$

where $\mu_t^i = \exp\left(-\eta[\widehat{L}_{t-1}^i + \ell(f_t^i, 0) + \widehat{H}_{t-1}^i]\right)$, $i \in [N]$.

Remark. Detailed proof is appended in the supplementary material. $\widehat{H}_{t-1}^i$ is an unknown loss suffered on the un-queried instances. We assume that all the instances are independently and identically distributed from some unknown distribution and the performance of one worker will not change over time. Given these two assumptions, we can easily verify that the cumulative loss of one worker on the unlabeled examples is propositional to that on the labeled examples. Formally, if there are m labeled examples and n unlabeled examples until time t, then we have $\mathbb{E}[\widehat{H}_{t-1}^i] = \frac{m}{n}\widehat{L}_{t-1}^i$.

Algorithm 1 summarizes the procedure of the proposed framework.

Algorithm 1 Budget Active Learning with Expert Advice

```
Input: a pool of experts $f^i,\ i = 1, \dots, N$.
Output: cumulative loss $\hat{L}^i, i = 1, \dots, N$.
Initialize tolerance threshold $\delta > 0, \beta > 0$ and $\widehat{L}_t^i = 0,\ i \in [N]$.
for $t = 1, \dots, T_1$ do
  receive $\mathbf{x}_t$;
  get annotations $f_t^i$, $i \in [N]$;
  decide who to label according to Equation (6);
  decide when to query $\mathbf{x}_t$ based on Equation (7) or Theorem 2 and 3;
  if $Z_t = 1$ then
    request label $y_t$;
    update $\widehat{L}_t^i = \widehat{L}_{t-1}^i + \ell(f_t^i, y_t),\ i \in [N]$;
  else
    skip the label request for instance $\mathbf{x}_t$
  end if
end for
```

IV. Experimental Results

A. *Baseline Algorithms*

Here we list all the algorithms used in the following experiments:

- MV: the Majority Voting algorithm which takes the majority label as the predicted label;
- SIGNAL: the classical Maximum Likelihood Estimation algorithm [8];
- BIAS: the state-of-the-art Multidimensional Wisdom of Crowds algorithm [25];
- EW: the regular Exponentially Weighted average forecaster algorithm [5];
- MAEW/WAEW: the Margin-based/Weighted difference of advices based Active EW algorithm based on Equation (7) or Theorem 2, and REW is the random version;
- EW-Who: the EW algorithm with strategy of Who to label based on Equation (6), and REW-Who is the random version;
- EW-All: the EW algorithm combining MAEW and EW-Who, and REW-All is the random version;
- GF: the regular Greedy Forecaster algorithm [5];
- MAGF/WAGF: the Margin-based/Weighted difference of annotations Active GF algorithm based on Equation (7) or Theorem 3 , and RGF is the random version;
- GF-Who: the GF algorithm with strategy of Who to label based on Equation (6), and RGF-Who is the random version;
- GF-All: the GF algorithm combining MAGF and GF-Who, and RGF-All is the random version;

B. *Experimental Testbed and Setup*

TABLE I. Datasets used in the following experiments.

Dataset	# Instances	# Train	# Test	# Workers	Type
fashion	4005	400	3605	6	real
fashion-cloth	4005	400	3605	6	real
a8a	32561	2604	23444	5	simulated
spambase	4601	368	3312	5	simulated
svmguide1	7089	567	5104	5	simulated

Table I shows the datasets used in our experiments. *fashion, fashion-cloth* are two real crowdsourcing datasets downloaded from UMassTrace [4] repository, and the other datasets are simulated datasets downloaded from LIBSVM [5] repository. To simulate the reliable workers in the simulated datasets, we adopt the following online learning algorithms implemented in the library LIBOL [13].

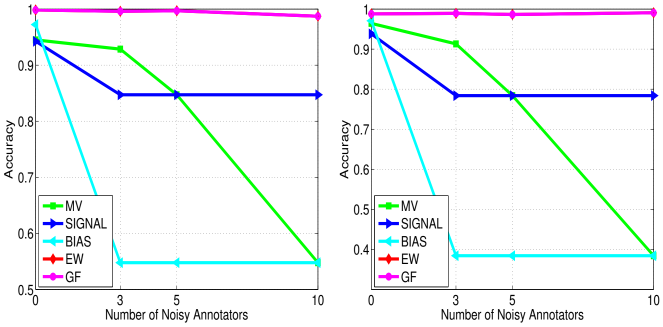
- PA: the Passive-Aggressive algorithm [6];
- AROW: the Adaptive Regularization Of Weights algorithm [7];
- ALMA$_p(\alpha)$: the Approximate Maximal Margin Algorithm [11];
- ROMMA: the Relaxed Online Maximum Margin Algorithm [17];
- PERCEPTRON: the classical Perceptron algorithm [19].

The parameter C of the PA algorithm is set to 5, and the parameter γ of the AROW algorithm is set to 1. For the ALMA$_p(\alpha)$ algorithm, p and α are set to 2 and 0.9, respectively. The learning rate η for Equation (3) and (4) is set to $\sqrt{8 \ln N/T}$ on all datasets, where N denotes the number of workers and $T = \#\ Train$ in Table I.

To investigate the performance of the algorithms in the scenarios where noisy workers exist, we manually added $\{0, 3, 5, 10\}$ noisy workers for each dataset. These four different number of noisy workers represent the scenarios where the number of noisy workers is much less than, less than, the same as and larger than the number of reliable workers, respectively. These four scenarios could almost simulate the situations happening in the real crowdsourcing platforms. Here the noise we considered is the group-cheating noise, where all the noisy workers added would give the same and random label for all the instances.

For each simulated dataset, 20% of the instances are used to train the five reliable workers. The rest of instances are divided into 10 parts. Each part of these 10 parts is used as standard questions to train the proposed algorithms, and the other 9 parts are used as test sets, where we use the workers' reliabilities learned on the 1 part. The accuracy (Y-axis) showed in the following figures is the average accuracy on the test sets among the 10 folds. Please notice, on the 9 test parts, true labels are not queried from domain experts and only their annotations and workers' reliabilities learned on the 1 part are used.

C. *Experiments on Different Noisy Scenarios*



(a) fashion

(b) fashion-cloth

Fig. 2. Accuracy V.S. Number of Group-cheating Workers.

In this section, we investigate the influence of noisy workers on baseline algorithms. Figure 2 shows the accuracy of different algorithms with different number of noisy workers. The performance on the other datasets is shown in supplementary material.

As expected, both the EW and GF algorithms are stable and outperform the state-of-the-art algorithms in all noisy levels. All of the other algorithms could perform as well as EW and GF in the noise-free scenario (0 noisy workers added) without any ground-truth labels requirement. This makes SIGNAL and BIAS algorithms attractive in large-scale annotation tasks, however, as the group-cheating noisy workers are added, all of the baseline algorithms greatly suffered as expected.

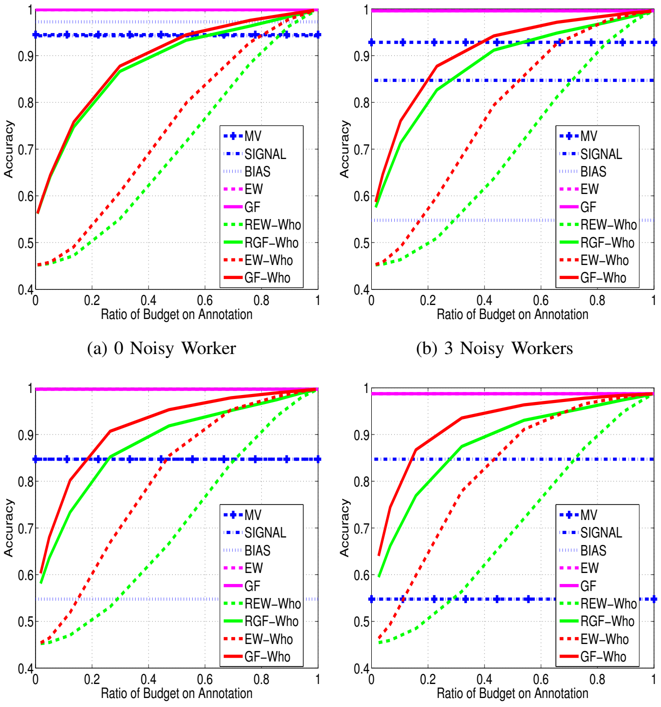
[4] http://traces.cs.umass.edu/index.php/Mmsys/Mmsys

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Especially for the MV algorithm, as the number of group-cheating workers increases, the annotations from noisy workers rapidly dominant the vote. This makes the MV algorithm infeasible. BIAS algorithm considers each workers bias based on the SIGNAL algorithm, this feature makes it outperform the other two baseline algorithms in the noise-free or random noisy setting [25], however, its performance also suffers most as group-cheating noisy workers are added. These observations motivate to consult domain experts to robustly learn workers' reliabilities in the group-cheating scenarios.

Although EW and GF algorithms could robustly achieve better performance, task allocation doesn't consider workers' reliabilities which could waste money on allocating task to noisy workers. What's worse, both EW and GF require all the golden labels on the training set, which is costly considering the high price to querying the domain experts. In the following subsections, we evaluate the proposed algorithms on tackling these two limitations, respectively.

D. Experiments on Who to Label



(a) 0 Noisy Worker

(b) 3 Noisy Workers

(c) 5 Noisy Workers

(d) 10 Noisy Workers

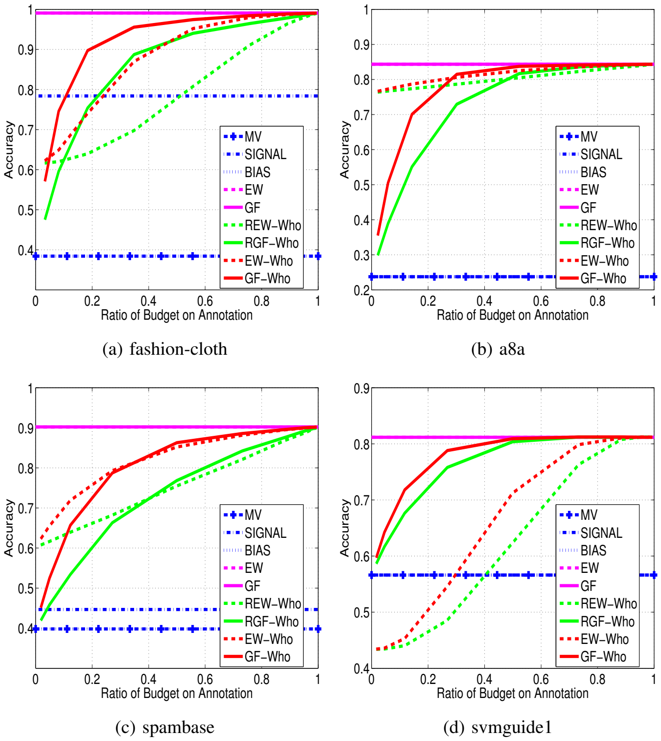
Fig. 3. Accuracy V.S. Budget Ratio in Different Group-cheating Scenarios on *fashion* Dataset.

In this section, we investigate the performance of the proposed active allocation strategy on "Who to Label" module in Figure 1. To simplify the evaluation, we assume each worker will spend 1 resource to label one instance, so the budget size equals the multiplication of # Train and # Workers shown in Table I. Figure 3 shows the performance with varied budget ratio under different noisy scenarios. The performance on the other datasets is shown in supplementary material.

Several observations could be made. First, on all noisy settings, the proposed active allocation algorithms, EW-Who and GF-Who, could consistently infer much more accurate labels than the random versions REW-Who and GF-Who, respectively. Specifically, In Figure 3 (a), where there is no extra noisy workers added, random algorithms REW-Who and RGF-Who could achieve similar performance as the active allocation algorithms EW-Who and GF-Who since the reliabilities of all workers are comparable, and there is no much difference of deciding who to label the given instance. However, as the number of noisy workers increases, shown in Figure 3 (b), (c), (d), the proposed reliability-based allocation algorithms based on Equation (6) greatly outperform their random versions. This confirms the effectiveness of considering the cumulative performance of each worker when assigning the annotation tasks.

Second, the proposed strategies could reduce the annotation budget to get comparable performance with baseline algorithms when noisy workers exist. For example, in Figure 3 (d), with less than %20 budget, GF-who could outperform all the other algorithms in terms of accuracy. This feature makes it much more attractive as the number of noisy workers increases.

Figure 4 shows the performance of proposed algorithms on the other datasets, where 10 extra noisy workers are added. Similar observations could be made. It should be noted that *BIAS* algorithm performs as bad as *MV* algorithm. Although the proposed active allocation algorithms could consistently achieve promising results, they are built on the workers' reliabilities which are learned by querying domain experts, who usually charge much higher than the noisy workers in the crowdsourcing platforms. In the following experiments, we investigate the proposed algorithms in reducing the number of queries of domain experts.



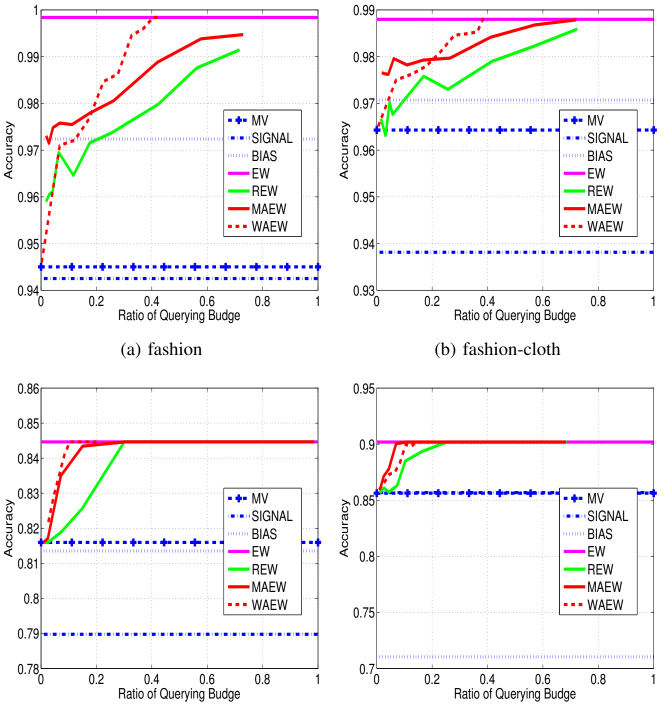
(a) fashion-cloth

(b) a8a

(c) spambase

(d) svmguide1

Fig. 4. Accuracy V.S. Budget Ratio with 10 Extra Noisy Workers Added on Four Different Datasets.



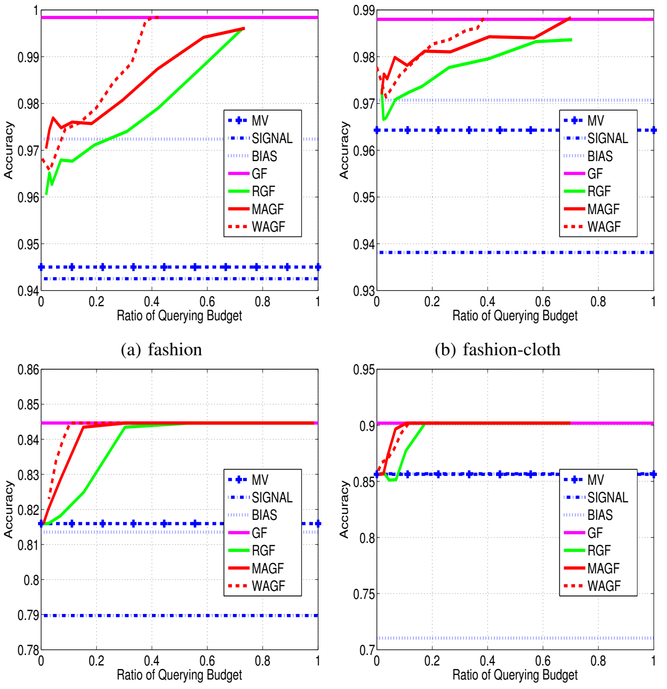
(a) fashion

(b) fashion-cloth

(c) a8a

(d) spambase

Fig. 5. Accuracy V.S. Budget Ratio with 0 Extra Noisy Workers Added Based on **EW** Forecaster.



(a) fashion

(b) fashion-cloth

(c) a8a

(d) spambase

Fig. 6. Accuracy V.S. Budget Ratio with 0 Extra Noisy Workers Added Based on **GF** Forecaster.

E. Experiments on When to Query

In this experiment, we assume the budgets equal "# Train" shown in Table I, and querying one instance for true label would cost one budget. We evaluate the accuracy on test set with respect to the querying ratio on training set, as shown in Figure 5 and 6. The performance on the other datasets is shown in supplementary material.

First, we could observe that the proposed algorithms (MAEW, MAGF, WAEW and WAGF) greatly outperform their random versions REW and RGF. Specially, the proposed WAEW and WAGF outperform the margin-based algorithms MAEW and MAGF once querying ratio is larger than a certain point. Besides, the MAEW and MAGF algorithms could achieve similar performance as EW and GF with limited querying budget, such as $\%40$ of querying budget in Figure 5 (a) (b). More importantly, with little extra cost in querying domain experts, the proposed algorithms are robust in accurately aggregating workers' annotations on different noisy scenarios.

Second, all active learning algorithms and their random versions could outperforms the MV algorithm with less than $\%10$ of querying budget (when the ratio of querying budget equals 0, the active learning algorithms are down to MV algorithm, where all workers are assigned with the same weight). These findings verify the effectiveness of the proposed active learning algorithms in reducing the cost of querying domain experts and achieving promising performance meanwhile.

F. Experiments on the Framework

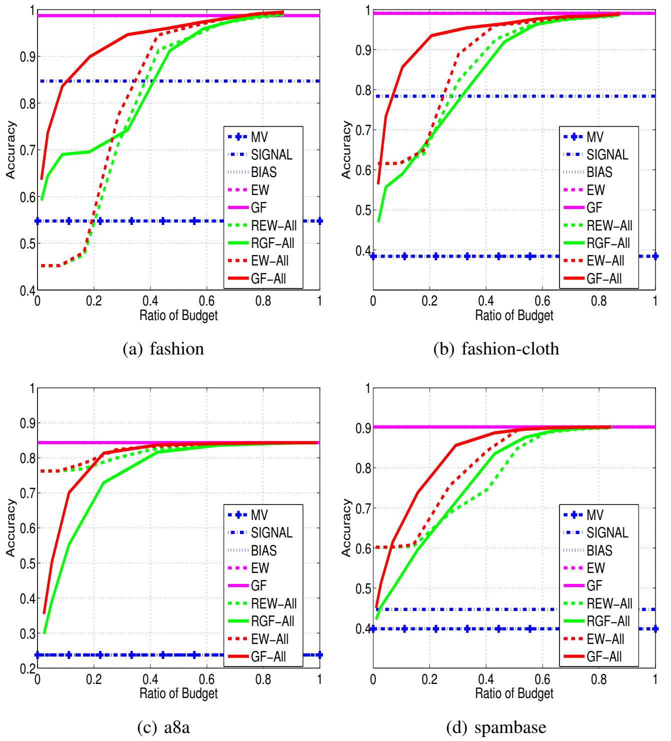
In this section, we investigate the overall performance of the proposed Framework 1 by combining the strategies of who to label (EW-Who and GF-Who) and strategies of when to query (MAEW and MAGF) together. To make a fair comparison with baseline algorithms, we assume the *total budget* as the number of annotations collected on "# Train" and "# Test" in Table I, and querying one instance with domain experts will cost m resources, where m equals the average number of annotations received from noisy workers for each instance. Ratio of budget equals the resources used on consulting domain experts with MAEW or MAGF algorithms and annotating with EW-Who or GF-Who algorithms over *total budget* defined.

Figure 7 shows the performance of the proposed framework with varied budget ratio on four datasets when 10 noisy workers are added. The performance on the other datasets is shown in supplementary material. Firstly, similar observations as in Figure 3 could be made. EW based algorithms achieve comparable performance as the one based on GF.

Secondly, by combining the "Who to Label" and "When to Query" strategies, the proposed framework could robustly infer much more accurate labels with limited budget. Although it needs domain experts to query the true labels, only 1% of the instances are queried. Besides, the learned reliabilities of workers could be directly and smoothly adopted in the allocation phrase. Considering all these advantages, the framework shows great potential to be applied into real annotation tasks, especially for the group-cheating scenarios.

V. Conclusion

In this article, we proposed a framework, budget active learning with expert advice, to infer accurate labels from noisy annotations. Specifically, we proposed two active learning strategies to decide when to query the ground-truth (which is



(a) fashion

(b) fashion-cloth

(c) a8a

(d) spambase

Fig. 7. Accuracy of Framework 1 V.S. Budget Ratio with 10 Extra Noisy Workers Added.

usually given by the domain experts) of an instance. Relying on the reliability of each worker learned, we proposed an active allocation strategy to decide who to label for an instance. We also carried extensive experiments both with real and simulated datasets in different scenarios. The empirical studies show the proposed active learning strategies could greatly alleviate intervention of domain experts, and the proposed active allocation strategy could effectively choose reliable workers to label. In summary, the proposed framework could achieve comparable results with the baseline algorithms in the normal setting, and robustly outperform the baselines in the group-cheating scenarios. In the future, we are interested in theoretically analyzing the proposed framework and evaluating its performance in interactive annotation tasks.

VI. Acknowledgement

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office. This research is also partially supported by Singapore MOE tier 1 research grant (C220/MSS14C003).

References

[1] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, Apr. 1988.

[2] D. C. Brabham. *Crowdsourcing*. Mit Press, 2013.

[3] A. Brew, D. Greene, and P. Cunningham. Using crowdsourcing and active learning to track sentiment in online media. In *ECAI*, pages 145–150, 2010.

[4] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear classification. *The Journal of Machine Learning Research*, 7:1205–1230, Dec. 2006.

[5] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[6] K. Crammer, O. Dekel, J. Keshet, and S. Shalev-shwartz. Online passive-aggressive algorithms. *The Journal of Machine Learning*, 7:551–585, 2006.

[7] K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. In *Advances in Neural Information Processing Systems (NIPS)*, pages 414–422, 2009.

[8] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[9] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[10] C. Eickhoff and A. P. de Vries. Increasing cheat robustness of crowdsourcing tasks. *Information retrieval*, 16(2):121–137, 2013.

[11] C. Gentile. A new approximate maximal margin classification algorithm. *The Journal of Machine Learning Research*, 2:213–242, 2002.

[12] J. Hannan. Approximation to bayes risk in repeated plays. *Contributions to the Theory of Games*, 3:97–139, 1957.

[13] S. C. Hoi, J. Wang, and P. Zhao. Libol: A library for online learning algorithms. *Journal of Machine Learning Research*, 15:495–499, 2014.

[14] D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[15] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[16] H. Li, B. Yu, and D. Zhou. Error rate analysis of labeling by crowdsourcing. In *ICML Workshop: Machine Learning Meets Crowdsourcing. Atalanta, Georgia, USA*, 2013.

[17] Y. Li and P. P. M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, pages 361–387, 2002.

[18] Q. Liu, A. Ihler, and M. Steyvers. Scoring workers in crowdsourcing: How many control questions are enough? In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1914–1922. 2013.

[19] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[20] N. Roy, A. Mccallum, and M. W. Com. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001.

[21] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52:55–66, 2010.

[22] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *14-th ACM SIGKDD*, pages 614–622, 2008.

[23] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23^{rd} international conference on World wide web*, pages 155–164. International World Wide Web Conferences Steering Committee, 2014.

[24] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *NIPS*, volume 10, pages 2424–2432, 2010.

[25] P. Welinder and P. Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *IEEE CVPRW*, pages 25–32, 2010.

[26] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, 2009.

[27] Y. Yan, G. M. Fung, R. Rosales, and J. G. Dy. Active learning from crowds. In *ICML*, pages 1161–1168, 2011.

[28] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcin. *arXiv preprint arXiv:1406.3824*, 2014.

[29] L. Zhao, G. Sukthankar, and R. Sukthankar. Robust active learning using crowdsourced annotations for activity recognition. In *Human Computation*, 2011.

[30] P. Zhao, S. Hoi, and J. Zhuang. Active learning with expert advice. *UAI*, pages 2–5, 2013.