Trust-Aware Optimal Crowdsourcing With Budget Constraint

Xiangyang Liu*, He He[†], John S. Baras[‡]
*[‡]Institute for Systems Research and Dept. of Electrical and Computer Engineering, University of Maryland College Park
[†]Dept. of Computer Science, University of Maryland College Park
Email: *xyliu@umd.edu, [†]hhe@cs.umd.edu, [‡]baras@umd.edu

Abstract—**Crowdsourcing has been extensively used for aggregating data from a large pool of workers. In a real crowdsourcing market, each answer obtained from a worker incurs cost. The cost is associated with both the level of trustworthiness of workers and the difficulty of tasks. Typically, access to expert-level (more trustworthy) workers is more expensive than to average crowd and completion of a challenging task is more costly than a click-away question. In this paper, we address the problem of optimal assignment of heterogeneous tasks to workers of varying trust levels with budget constraint. Specifically, we design a trust-aware task allocation algorithm that takes as inputs the estimated trust of workers and pre-set budget, and outputs the optimal assignment of tasks to workers. We derive the bound of total error probability that relates to budget, trustworthiness of crowds, and costs of obtaining labels from crowds naturally. Higher budget, more trustworthy crowds, and less costly jobs result in lower theoretical bound. Our allocation scheme does not depend on the specific design of the trust evaluation component. Therefore, it can be combined with generic trust evaluation algorithms. Our algorithm outperforms state-of-the-art by up to 30% on real data.**

I. Introduction

Crowdsourcing provides a convenient and efficient way for data collection without having to acquire costly labels from domain experts. In a typical crowdsourcing task, a requester distributes small jobs to non-expert workers and provides a small amount of payment upon job completion. Such a small job can be translating a sentence [1], annotating an image [2], classifying search queries [3], etc. Answers (or labels) obtained from workers are usually noisy due to workers' lack of expertise, carelessness, or malicious labeling. To mitigate the noise, one question is redundantly distributed to multiple workers and the answers are aggregated to produce a single answer, expected to be more accurate. Many crowdsourcing platforms are available, for example, Amazon Mechanical Turk, ESP game and reCaptcha.

One typical goal in crowdsourcing tasks is to infer the ground truth from collected answers. Much work [4], [5], [6] in crowdsourcing has been devoted to making aggregated decisions to predict true labels given noisy and even malicious input from workers. However, these algorithms do not consider the cost incurred from obtaining a label from a worker; while in practice, the number of answers we can get is restricted by the budget coming with requesters. Under this constraint, a natural question to ask is how to allocate tasks to workers adaptively with limited budget.

Past approaches to crowdsourcing with budget constraint have assumed that all questions and workers are homogeneous – questions do not differ in difficulty level, and all workers are as capable as each other and get the same payment for answering any question. This can be an over-simplified setting for real problems. In practice, the cost depends on both the question and the worker. For example, fine category classification of different kinds of birds requires more domain knowledge than simply telling if there is a bird in an image; summarizing a paragraph needs more work than deciding if a tweet is positive or negative. Requesters generally pay more to workers for difficult tasks. On the other hand, skillful workers ask for higher payment than ordinary workers, and have a larger chance of providing the ground truth. For example, for the same task, consulting a domain expert is more costly than asking a random worker on Mechanical Turk; however, on average more Turks are required to infer the correct answer. Thus there is a trade-off between cost and answer quality. A more cost-efficient way to task distribution than blind random assignment would be to assign easy tasks to cheap workers and hard tasks to workers with more expertise. The answers given by workers are then combined with estimated trustworthiness of workers. We consider the trustworthiness of a worker as equivalent to the worker's reliability. Therefore use trust and reliability interchangably in this paper. Specifically, expert level crowd has higher trust value while common non-expert crowd has lower trust value.

In this paper, we address the problem of trust-aware task allocation by considering cost and expertise variation among workers. We propose an easy-to-implement allocation algorithm in the setting of weighted majority vote with theoretical guarantee. We formulate the assignment problem as a nonlinear integer programming problem with budget constraint, and relax it to a convex optimization problem that has an analytical solution. We also give a theoretical error bound to our algorithm.

Our contributions are as follows:

- We formalize the problem of trust-aware task allocation in crowdsourcing and provide a principled way to solve it.
- Our formulation models the workers' trustworthiness and the costs depend on both the question and the worker group. Our method is ready to be extended to more complicated aggregation method other than the weighted majority vote as well.
- The trust-aware task allocation scheme we propose can achieve $\frac{N}{2} - \mathcal{O}(\sqrt{B})$ for total error probabilities, where N is the number of tasks and B is the total budget. Different from [7], the exact performance

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

bound of error probability also incorporates both trustworthiness of crowds and cost. More trustworthy crowds and less costly jobs result in lower guaranteed bound.

II. Related Work

Most previous works focus on aggregating labels from multiple workers. None of them address a practical issue: the job requester has a budget constraint and he wants to make the best use of the requester's budget. A closely related work along this line is Crowdscreen [8] that developed algorithms for minimizing expected cost regarding number of questions asked and the estimation accuracy. However, the cost of assigning different questions is assumed to be uniform and the heuristics-based algorithms have no theoretical guarantee of performance. This guarantee is given in [9] where all questions are homogeneous and the upper bound they derived is valid only when the number of questions assigned approaches infinity, rendering it impractical. Works that further investigate the problem of task assignment for heterogeneous tasks include [10], [7]. The former is focused on minimizing cost subject to a quality constraint when workers arrive online while the latter is in the direction of minimizing estimation error under budget constraint and the cost associated with questions varies w.r.t difficulty. In particular, in [7], cost is determined by only the difficulty of questions and they can not choose explicitly which experts to choose for the completion of the task.

III. Problem Setting

We consider classification tasks where the wisdom of crowds is utilized to estimate the ground truth of each instance. We assume that there are N tasks and the difficulty of task i can be mapped to a real number d_i. We consider binary classification and denote the unknown true label of task i by $r_i \in \{-1, 1\}$. However, our algorithm can be applied to general classification tasks as well. We further assume that there are M crowds available for the job requesters. As can be expected, in real life, some crowds behave professionally and provide reliable answers, while other crowds are not as trustworthy, either because they have lower expertise level or because they want to get the payment without investing enough effort. We denote the answer given by a worker k from crowd j for task i as $\ell_{j_k i} \in \{-1, 1\}$. The job requester comes to the crowdsourcing market with a fixed budget B and he/she expects to get the highest performance out of the given budget. The crowdsourcing platform has a scheduler that distributes tasks to its pool of workers. Each assignment of task i to a worker from crowd j is associated with a cost c_{ij}.

We adopt a 1-coin model to describe the worker's stochastic behavior when answering a specific question. The 1-coin model assumes the probability of labeling a question with 1 given $r_i = 0$ equals the probability of labeling it with 0 given $r_i = 1$. We denote the probability of getting a correct answer of task i given by worker k from crowd j by u_{ij_k}. A higher value of u_{ij_k} indicates higher trust value. Extension of our work to a 2-coin model (a worker is modeled by two parameters when the truth label is binary, i.e. the probability of giving correct label when truth label is 0 and the probability of giving correct label when truth label is 1) is straightforward. Given the symmetry present in the definition of the 1-coin model, without loss of generality, we assume that the true label r_i of task i is 1. Thus the answer given by worker k from crowd j follows the Bernoulli distribution: $\ell_{ij_k} \sim \text{Bin}(1, u_{ij_k})$. For each task i, a user from crowd j is sampled according to some unknown distribution and we denote the expected trust value of crowd j toward task i $\mathbb{E}[u_{ij_k}]$ as u_{ij}. Note that the random variables u_{ij} and u_{ij_k} are unknown.

We assume that there is a separate trust evaluation component that assesses each worker's trustworthiness and outputs estimates of a crowd's trust value, denoted by $w_j \in [0, 1]$, which represents the trust evaluation component's belief about the probability that workers from crowd j's answer a question correctly. We choose to estimate the trust value of a whole crowd instead of individual workers. In reality, companies like CrowdFlower[1] provides hierarchies of workers ranging from domain experts to average open crowd, thus it is more reasonable to keep track of the performance of each crowd than that of individuals.

A common approach to ground truth inference in crowdsourcing is weighted majority vote:

$$\hat{r}_i = \text{sign}\left(\sum_{j=1}^{M}\sum_{k=1}^{n_{ij}} w_j \ell_{j_k i}\right) \tag{1}$$

where w_j is the estimated trust value of crowd j, $\ell_{j_k i}$ is the answer to question i provided by worker j who belongs to crowd j, and n_{ij} is the number of workers from crowd j allocated for question i. The above estimation is a very basic algorithm in crowdsourcing and is usually used as the baseline or a preprocessing step for more sophisticated methods. Therefore, we use the error probability based on the weighted majority vote as an upper bound of the error probability we can achieve. Given the fixed budget provided by the job requester, the scheduler has two options. It either assigns a set of budget constraints B_i for each task i since we don't want to allocate all the budget to a single question or the scheduler just has a budget constraint on the total expense for completing all the tasks. For each task i, multiple workers are assigned to provide answers for it. The number of workers from crowd j assigned to task i is denoted by n_{ij} and the set of workers assigned to task i can be compactly expressed as $n_i = \{n_{ij}\}_{j=1}^{M}$. In the setting of fixed total budget across all tasks, the optimal crowdsourcing problem becomes:

$$\begin{aligned} \underset{n_{ij}}{\text{minimize}} \quad & \sum_{i=1}^{N} \Pr\left(\hat{r}_i\left(\{n_{ij}\}_{j=1}^{M}, w\right) \neq r_i\right) \\ \text{subject to} \quad & \sum_{i,j} c_{ij} n_{ij} \leq B \\ & n_{ij} \in \mathbb{N} \end{aligned} \tag{2}$$

which is generally a non-deterministic nonlinear integer programming problem. When we substitute question i's true label r_i with the estimated label $\hat{r}_i$ using the weighted majority vote equation (1), equation (2) is relaxed.

There is a trust evaluation component that gives estimation of crowds' trustworthiness w_j. Note that sometimes we might need trustworthiness of a crowd with respect to different types

[1] http://www.crowdflower.com/

of questions, which is questions of varying difficulty in our case. For simplicity, in algorithm 1 and algorithm 2 that follow in Section IV, just a scalar parameter w_j is assumed for each crowd. Extension to trustworthiness with respect to each type of questions is straighforward. The design of the trust evaluation algorithm is beyond the scope of this paper. Interested readers are referred to [11], [12] for basics on trust models. We assume we can get access to the estimation of trustworthiness given by this component and our allocation scheme goes from there. Our allocation scheme works with a general trust estimation component. Note that trust estimation is usually not given and incurs further cost. However, practical crowdsourcing platforms use a pipeline model, where separate components are dedicated to trust estimation, task allocation and answer inference. We intend to keep our job (task allocation) as independent from others as possible, yet flexible enough to join with any algorithm of other components. The output of our allocation scheme is a set of assignments n_{ij}. Note that we are considering task assignment before tasks are deployed in the crowdsourcing market, i.e., trust values are static in this case. This is justified by the observation that most crowdsourcing marketplaces like Amazon Mechanical Turk require preset numbers of workers to questions before deployment. That said, given time-varying trust estimates, our method can be easily made online – do partial assignment, wait for answers, update trust estimates and do another batch of assignment.

IV. Trust-Aware Task Allocation

Our proposed budget allocation strategy is trust-aware in the sense that it utilizes the estimated trustworthiness of crowds given by trust evaluation component and allocation decision is partially influenced by the estimation. The process works as follows. We present the optimal budget allocation scheme with total budget constraint. The job allocator selectively assigns multiple workers from each crowd j to each task i given the estimated trustworthiness w_j and cost c_{ij}.

A. Assumptions

For question i, we assume that the user k from crowd j samples his/her answer $\ell_{j_k i}$ from a Bernoulli distribution, i.e., $\ell_{j_k i} \sim \text{Bin}(1, u_{ij_k})$. The expected answer $\mathbb{E}_{\text{Bin}(1,u_{ij_k})}[\ell_{j_k i}]$ is μ_{ij_k}. We assume that a user k is picked from a crowd j uniformly and $\mathbb{E}_{k\sim U_j}[\mu_{ij_k}] = \mu_{ij}$, where μ_{ij} denotes expected trust value of crowd j. For an allocation $\{n_{ij}\}, i = 1, \ldots, N, j = 1, \ldots, M$, we define the expected answer for question i averaged over workers from crowd j as

$$\mu_i = \frac{\sum_{j=1}^{M} n_{ij} w_j \mu_{ij}}{\sum_{j=1}^{M} n_{ij}} = \sum_{j=1}^{M} \rho_{ij} w_j \mu_{ij} \tag{3}$$

where $\rho_{ij} = \frac{n_{ij}}{\sum_{j=1}^{M} n_{ij}}$ and is fully determined by the allocation $\{n_{ij}\}$. We assume that the weighted majority voting aggregating scheme yields a somewhat reasonable performance for the given task i under uniform allocation, i.e. $\rho_{ij} = \rho_i, \forall j$:

$$\begin{cases} \mu_i \geq 0 & \text{if}\, r_i = 1 \\ \mu_i < 0 & \text{if}\, r_i = -1 \end{cases} \tag{4}$$

This means that if our assignment for question i is at least as good as uniformly random assignment, the expected answer for question i in equation (3) has the same sign as the ground truth.

B. Optimization Problem

Let $Y_i = \sum_{j=1}^{M} \sum_{k=1}^{n_{ij}} w_j \ell_{j_k i}$. The error probability of task i in equation (2) can be relaxed by using the Hoeffding concentration bound:

$$\Pr\left(\hat{r}_i \neq r_i\right) \leq \exp\left(- \frac{\left(\sum_{j=1}^{M} n_{ij} w_j (2u_{ij} - 1)\right)^2}{2\sum_{j=1}^{M} n_{ij} w_j^2} \right) \tag{5}$$

where u_{ij} denotes the expected trust value of crowd j and w_j denotes the estimated trust value for crowd j. equation (5) makes the problem in equation (2) a deterministic optimization problem. However, this is not convex in general.

Next we discuss how to relax the deterministic objective function on the right hand side of equation (5) by probably approximately correct learning framework (PAC) [13]. We consider the situation where the actual obtained answer deviates from the expected answer by ϵ_i. Using the Hoeffding Inequality, we can get

$$\Pr\left(\left| \frac{1}{\sum_{j=1}^{M} n_{ij}} \sum_{j=1}^{M} \sum_{k=1}^{n_{ij}} w_j \ell_{j_k i} - \mu_i \right| \geq \epsilon_i \right) \leq$$
$$2\exp\left\{ -\frac{\epsilon_i^2 (\sum_{j=1}^{M} n_{ij})^2}{2\sum_{j=1}^{M} n_{ij} w_j^2} \right\}$$

Now let $2\exp\left\{ -\frac{\epsilon_i^2 (\sum_{j=1}^{M} n_{ij})^2}{2\sum_{j=1}^{M} n_{ij} w_j^2} \right\} = \beta$, where β is a chosen real number from 0 to 1. This means that with probability at least $(1-\beta)^N$, the following holds: $\hat{r}_i \in [\mu_i - \epsilon_i, \mu_i + \epsilon_i]\ \forall i$. We express ϵ_i as:

$$\epsilon_i = \sqrt{\frac{-2\ln\frac{\beta}{2} \sum_{j=1}^{M} n_{ij} w_j^2}{\left(\sum_{j=1}^{M} n_{ij}\right)^2}} \tag{6}$$

In practice, the value of β depends on the required confidence level. Usually β is small, thus the above interval for $\hat{r}_i$ is of high probability. In the following argument we will only consider the case where $\hat{r}_i$ lies in the interval $[\mu_i - \epsilon_i, \mu_i + \epsilon_i]$. If $\mu_i - \epsilon_i \geq 0$, then $\hat{r}_i = \text{sign}\left(\sum_{j=1}^{M} \sum_{k=1}^{n_{ij}} w_j \ell_{j_k i}\right) \geq 0$, the answer will always be correct. If $\mu_i - \epsilon_i < 0$, then we will get the wrong answer with probability $\frac{\epsilon_i - \mu_i}{2\epsilon_i}$. Therefore, in the interval we are considering, we have

$$\Pr(\hat{r}_i \neq r_i) = \max\{0, \frac{\epsilon_i - \mu_i}{\epsilon_i}\} \leq \frac{1}{2} - \mu_{\min}\frac{1}{2\epsilon_i} \tag{7}$$

where $\mu_{\min} = \min \mu_{ij}$.

We would like to minimize the error probability summing over the N tasks, and our optimization objective is

$$\underset{n_{ij}}{\text{minimize}} \; -\sum_{i=1}^{N} \sqrt{\frac{\left(\sum_{j=1}^{M} n_{ij}\right)^2}{\sum_{j=1}^{M} n_{ij} w_j^2}}$$

This is not necessarily a convex function, however, notice that $\sum_{j=1}^{M} n_{ij} \geq \sum_{j=1}^{M} n_{ij} w_j^2$ since $w_j \in [0, 1]$, and we can relax $-\sum_{i=1}^{N}\sqrt{\frac{\left(\sum_{j=1}^{M} n_{ij}\right)^2}{\sum_{j=1}^{M} n_{ij}w_j^2}}$ to its upperbound $-\sum_{i=1}^{N}\sqrt{\frac{\left(\sum_{j=1}^{M} n_{ij}w_j^2\right)^2}{\sum_{j=1}^{M} n_{ij}w_j^2}}$. This relaxation results in a convex optimization problem given by:

$$\begin{aligned} \underset{n_{ij}}{\text{minimize}} \quad & -\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{M} n_{ij}w_j^2} \\ \text{subject to} \quad & \sum_{ij} c_{ij}n_{ij} \le B \\ & n_{ij} \ge 0,\ i = 1, \ldots, N, j = 1, \ldots, M \end{aligned} \tag{8}$$

The optimal solution for the above problem can be expressed as:

$$n_{ij} = \begin{cases} \frac{B}{\frac{c_{ij_i^*}^2}{w_{j_i^*}^2}\sum_{l=1}^{N}\frac{w_{j_l^*}^2}{c_{lj_l^*}}} & \text{if } j = j_i^* \\ 0 & \text{if } j \ne j_i^* \end{cases}, \tag{9}$$

$$\text{where } j_i^* = \arg\max_j \frac{w_j^2}{c_{ij}} \text{ and } i = 1, 2, \ldots, N.$$

From the optimal allocation scheme we can see that our model prefers the most cost-efficient crowd in terms of the ratio of its level of trust over cost. Since n_{ij} might be fractional, we set it to be $\lfloor n_{ij} \rfloor$. The full algorithm is shown in Algorithm 1 and we call it TAA for short.

Algorithm 1: Trust-Aware Assignment

```
Input: N tasks, budget B, worker cost
       $c_{ij}(i = 1, \ldots, N, j = 1, \ldots, M$
Output: job allocations $n_{ij}$, predicted answer $\hat{r}_i$
$Br = B$;
for $i = 1 : N$ do
    $j_i^* = \arg\max_j \frac{w_j^2}{c_{ij}}$
    for $j = 1 : M$ do
        if $j = j_i^*$ then
            $n_{ij} = \left\lfloor \frac{B}{\frac{c_{ij_i^*}^2}{w_{j_i^*}^2}\sum_{l=1}^{N}\frac{w_{j_l^*}^2}{c_{lj_l^*}}} \right\rfloor$
            $Br \leftarrow Br - \sum_{j=1}^{M} n_{ij_{i^*}}c_{ij_{i^*}}$
        else
            $n_{ij} = 0$
        end
    end
end
while $Br > 0$ and $i \le N$ do
    $n_{ij_{i^*}} = n_{ij_{i^*}} + 1$, $Br = Br - c_{ij_{i^*}}$, $i = i + 1$
end
Use weighted majority voting to estimate answers
```

The solution in equation (9) exhibits sparsity features since: 1) for any question, budget is allocated to only one of the crowds; and 2) when taking the floor, difficult questions tend to get 0 budget while easy questions get the whole share of the budget. We propose to address this problem by introducing an extra regularization term which penalizes the sparse behavior of allocation in Algorithm 1. For the sake of convenience, we relax the objective function in equation (8) by $-\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{M} n_{ij}w_j^2} \ge -\sum_{i=1}^{N}\sum_{j=1}^{M} n_{ij}w_j^2$. Therefore the optimization problem becomes:

$$\begin{aligned} \underset{n_{ij}}{\text{minimize}} \quad & -\sum_{i=1}^{N}\sum_{j=1}^{M} n_{ij}w_j^2 + \frac{\xi}{2}\|n\|_2^2 \\ \text{subject to} \quad & \sum_{ij} c_{ij}n_{ij} \le B \\ & n_{ij} \ge 0,\ i = 1, \ldots, N, j = 1, \ldots, M \end{aligned} \tag{10}$$

The optimal solution of this problem is

$$n_{ij} = \frac{1}{\xi}\left(w_j^2 - c_{ij}\frac{\sum_{kl} c_{kl}w_l^2}{\sum_{kl} c_{kl}^2}\right) + \frac{Bc_{ij}}{\sum_{kl} c_{kl}^2}, \forall i, j$$

where ξ should be chosen such that n_{ij} is positive. We can see from this solution structure that for each question, budget will be allocated to multiple crowds instead of just one. The penalty term in equation (10) gives credits to allocations that are more spread out, which makes the bound closer to equation (8). The full algorithm is shown in Algorithm 2 and we call it TAAP for short.

Algorithm 2: Trust-Aware Assignment With Penalty

```
Input: $N$ tasks, budget $B$, worker cost
       $c_{ij}(i = 1, \ldots, N, j = 1, \ldots, M$
Output: job allocations $n_{ij}$, predicted answer $\hat{r}_i$
$Br = B$;
for $i = 1 : N$ do
    for $j = 1 : M$ do
        $n_{ij} = \left\lfloor \frac{1}{\xi}\left(w_j^2 - c_{ij}\frac{\sum_{kl} c_{kl}w_l^2}{\sum_{kl} c_{kl}^2}\right) + \frac{Bc_{ij}}{\sum_{kl} c_{kl}^2} \right\rfloor$
        $Br \leftarrow Br - \sum_{j=1}^{M} n_{ij}c_{ij}$
    end
end
while $Br > 0$ do
    for $i = 1 : N$ do
        if $Br > 0$ then
            Randomly choose $j$th crowd
            $n_{ij} = n_{ij} + 1$, $Br = Br - c_{ij}$
        else
            Break
        end
    end
end
Use weighted majority voting to estimate answers
```

V. Theoretical Performance Bound

In this section, we discuss the performance of the allocation solution given by our proposed trust-aware allocation by providing the guaranteed upper bound of the error probability of the original optimization problem of equation (2).

Theorem 1. *For any* $\sum_{ij} c_{ij}n_{ij} \le B$, *the total error probability is less than or equal to* $\sum_{i=1}^{N}\exp\left(-\frac{Bw_{j_i^*}^2\left(2u_{ij_i^*}-1\right)^2}{2c_{ij_i^*}^2\sum_{l=1}^{N}\frac{w_{j_l^*}^2}{c_{lj_l^*}}}\right)$, *where* $j_i^* = \arg\max_j \frac{w_j^2}{c_{ij}}$.

This result is intuitive in that the larger the budget we have, the lower the error probability bound we can obtain. The bound improves exponentially with respect to budget increase. In addition, lower cost of c_{ij} and higher trust value $u_{ij_i^*}$ lead to lower error bound.

We can actually obtain an improved upper bound that holds with high probability from the perspective of PAC, like the work in [7].

Theorem 2. *For any $\sum_{j=1} c_{ij} n_{ij} \leq B$, the total error probability satisfies:*

$$\sum_{i=1}^{N} \Pr\left(\hat{r}_i \neq r_i\right) \leq$$

$$\max\left\{0, \frac{N}{2} - \sum_{i=1}^{N} \mu_{min} \sqrt{\frac{1}{-8\ln\frac{\beta}{2}} \frac{Bw_{j_i^*}^4}{c_{ij_i^*}^2 \sum_{l=1}^{N} \frac{w_{j_l^*}^2}{c_{lj_l^*}}}}\right\} \tag{11}$$

, where $j_i^* = \arg\max_j \frac{w_j^2}{c_{ij}}$.

VI. Experimental Results

Besides the theoretical results given in Section V, we also evaluate the performance of our proposed *trust-aware assignment* (TAA) and *trust-aware assignment with penalty* (TAAP) on a real dataset and compared them against benchmark algorithms such as *uniform assignment* (UA) and algorithms from [7] adjusted to our setting, i.e. *crowd-quality-seeking assignment* (CQSA) and *cheap assignment* (CA). We show that our algorithms outperform state-of-the-art.

A. Benchmark Algorithms

The set of benchmark algorithms we use for comparison are:

1) UA: the algorithm tends to allocate the same number of people to answer a question from each available crowd. If the budget is not used up, for each question, it randomly chooses an expert from the set of crowds.
2) CQSA: for each question, the algorithm only chooses people from the most trustworthy crowd to assign to that question according to $n_{ij_i} = \left\lfloor \frac{B}{c_{ij_i}^2 \sum_{i=1}^{N} \frac{1}{c_{ij_i}}} \right\rfloor$, where $j_i = \arg\max_j w_j$ If budget is not consumed, it iterates the question set again and randomly chooses an expert from the set of crowds for each question.
3) CA: the algorithm only chooses the cheapest crowd (the least trustworthy crowd) for questions according to $n_{ij_i} = \left\lfloor \frac{B}{c_{ij_i}^2 \sum_{i=1}^{N} \frac{1}{c_{ij_i}}} \right\rfloor$, where $j_i = \arg\min_j w_j$. The same procedure is done as in crowd-quality-seeking assignment when budget is not used up.

After the assignment stage, weighted majority vote, as in equation (1), is applied to the algorithms.

B. Experiment Setup on Galaxy Zoo Dataset

The real dataset we use is Galaxy Zoo [14], a set of galaxy annotations contributed by a crowd of volunteers who are non-experts. The dataset contains statistics about votes of these volunteers for over 900,000 galaxies. The images of these galaxies are classified as elliptical (E), combined spiral (CS), or unknown by volunteers. The dataset from Galaxy Zoo used in our paper is SDSS image release 7. A subset of 700 galaxies that are classified as class elliptical or combined spiral is randomly chosen. These classified galaxies have more than 80% agreement and the class agreed upon can be treated as truth label.

Classification of galaxy images from Galaxy Zoo does not have explicit difficulty levels and volunteers that participate in giving classifications do not have explicit level of trust either. However, we first divide the 700 galaxies into 2 groups based on the level of agreement. The first group is considered easy questions and the second group is considered difficult questions. The level of agreement in the first group is higher than that in the second. Then we simulate three kinds of crowds with increasing level of trustworthiness. Let α_t denote the difficulty parameter of type t question and β_j denote the trust parameter of type crowd j. Specifically α_t is scaled to $[5.0, 1.0]$ for easy and difficult questions respectively and the β scaled to $[0.65, 0.85, 0.98]$. Then we choose the trust value of crowd j toward question i as a sigmoid function of α_{t_i} and β_j: $u_{ij} = \frac{1}{1+\exp\left(-\alpha_{t_i}\beta_j\right)}$, where t_i is the type of the ith question, which is easy or difficult in our case. Next we assume the input from the trust evaluation component is $w_{ij} = 2u_{ij} - 1$. In practice, this might not be the case. However, any good design of trust evaluation algorithm should output higher trust value for more reliable crowd and lower trust value for less reliable crowd and the assumption that $w_{ij} = 2u_{ij} - 1$ also exhibits such behavior.

With these models, we choose the cost function that maps the question difficulty and crowd's trust value to money in the following way: for easy questions, the cost of different crowds is $[0.1, 0.5, 0.9]$ and the cost for difficult questions is $[0.3, 0.6, 1.0]$. The cost function along with the trustworthiness values captures the following intuitive ideas: 1) for each question type, more trustworthy crowd incurs higher cost; and 2) for a particular crowd, answering difficult questions incurs higher cost than answering easy ones.

C. Analysis

To test the performance of our proposed algorithm, we plot the total probability error as the budget increases from 50 to 1500. The result is depicted in Fig. 1. It is easy to see that our proposed TAAP outperforms all other algorithms across the span of budget. In particular, when the budget is relative small, i.e. $B \leq 200$, both TAA and TAAP improve over CQSA and UA by up to 30%. This indicates that our algorithms excel in efficient allocation when budget is not abundant. Also, the cheap assignment algorithm does equally well when budget is small since there is not enough budget for answering difficult questions and people from a cheap crowd can answer easy questions equally well compared to an expensive crowd. When the budget is abundant, however, TAA behaves poorly compared to other algorithms except for CA. This is due to

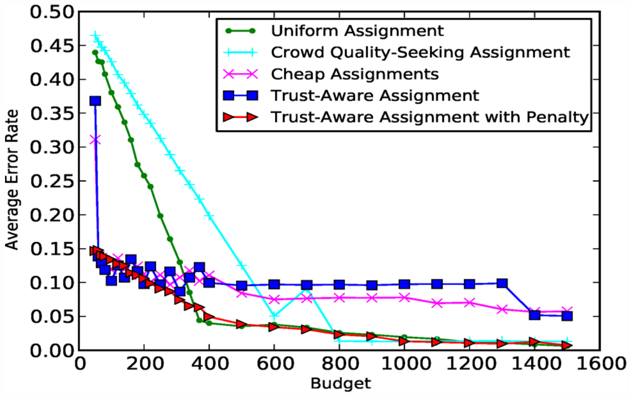

Fig. 1. Total error probability of algorithms UA CQSA CA TAA TAAPon Galaxy Zoo dataset with budget ranging from 50 to 1500.

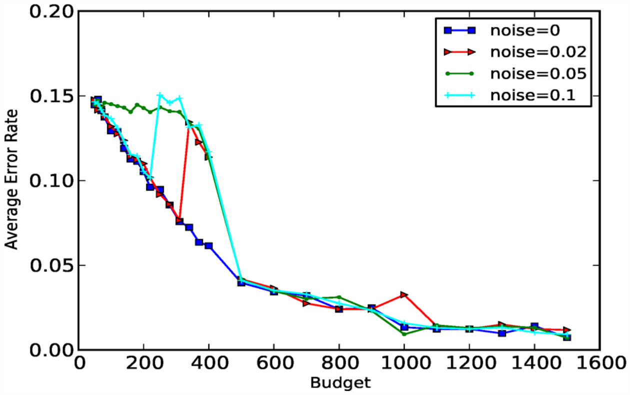

Fig. 2. Total error probability of algorithm TAAP on Galaxy Zoo dataset under noise variance from 0 to 0.1 and the budget is from 50 to 1500.

two reasons: 1) taking the floor in equation (9) makes many of the assignments 0, greatly deteriorating performance; and 2) the sparsity feature of equation (9), as mentioned earlier, did not switch to most trustworthy crowd even if budget is very high. TAAP addresses this problem and we can see that when budget is high, the algorithm does equally well compared to CQSA and UA.

The result in Fig. 1 assumes the trustworthiness can be perfectly estimated. Next we investigate the performance of TAAP when w_{ij} can not be perfectly estimated. By adding a Gaussian noise ϵ to u_{ij}, we have $w_{ij} = 2\left(u_{ij} + \epsilon\right) - 1$. We test TAAP with increasing variance of the noise ϵ ranging from 0 (perfectly estimated) to 0.1. Since u_{ij} takes value from $[0.5, 1]$ in our case, 0.1 is a significant noise variance. In Fig. 2, when budget is low, TAAP is to some extend affected by increasing noise variance. However, the error rate never increases by more than 5%, which is acceptable. When budget is sufficient, the algorithm is robust to varying noise variance levels and performs equally well compared to the case when trust can be perfectly estimated.

VII. Conclusion

In this paper, we considered the practical problem of budget allocation with trust estimation of different crowds. We would like to maximize the prediction accuracy within a given budget. In our setting, costs depend on both the question and the crowds grouped by level of expertise. We relaxed this accuracy-cost trade-off problem to a convex optimization problem by a PAC bound. We showed that there is a simple and intuitive closed-form solution to the convex problem. TAA always selects the most cost-efficient group for a given question and has at most $\frac{N}{2} - \mathcal{O}\left(\sqrt{B}\right)$ prediction error. In addition, to address the problem of flooring and sparsity feature exhibited in TAA, we proposed TAAP and showed its outstanding performance through experiments on a real dataset across budget span.

Note that though we experimentally investigated the effect of trustworthiness estimation error, we did not theoretically explore the effect of it on the total error probability. We plan to further analyze this in the future. Additionally, the truth label in this paper is assumed to be discrete and binary. We also would like to investigate the continuous values.

VIII. Acknowledgements

Research partially supported by grants AFOSR MURI FA-9550-10-1-0573, NSF CNS-1035655, NSF CNS-1018346 and by Maryland Procurement Office contract H98230-14-C-0127.

References

[1] O. F. Zaidan and C. Callison-BurchRichard, “Crowdsourcing translation: Professional quality from non-professionals,” 2011.

[2] C. Wah, S. Branson, P. Perona, and S. Belongie, “Multiclass recognition and part localization with humans in the loop,” 2011.

[3] R. M. C. McCreadie, C. Macdonald, and I. Ounis, 2010.

[4] R. Snow, B. O’Connor, D. Jurafsky, and A. Y. Ng, “Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks,” in *Empirical Methods on Natural Language Processing (EMNLP)*, 2008, pp. 254–263.

[5] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan, “Whose vote should count more: Optimal integration of labels from labelers of unknown expertise,” in *Advances in Neural Information Processing Systems*, 2009, pp. 1207–1216.

[6] P. Welinder, S. Branson, S. Belongie, and P. Perona, “The multidimensional wisdom of crowds,” in *Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 2424–2432.

[7] L. Tran-Thanh, M. Venanzi, A. Rogers, and N. R. Jennings, “Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks,” in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.

[8] A. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom, “Crowdscreen: Algorithms for itering data with humans,” in *International Conference on Management of Data (SIGMOD)*, 2012.

[9] D. Karger, S. Oh, and D. Shah, “Iterative learning for reliable crowdsourcing systems. neural information processing systems,” in *Neural Information Processing Systems (NIPS)*, 2011.

[10] C. Ho, S. Jabbari, and J. W. Vaughan, “Adaptive task assignment for crowdsourced classification,” in *International Conference on Machine Learning (ICML)*, 2013.

[11] G. Theodorakopoulos and J. Baras, “On trust models and trust evaluation metrics for ad hoc networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 318–328, 2006.

[12] A. Jsang and R. Ismail, “The beta reputation system,” in *Proceedings of the 15th bled electronic commerce conference*, 2002, pp. 41–55.

[13] L. G. Valiant, “A theory of the learnable,” *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[14] C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land *et al.*, “Galaxy Zoo 1 : Data Release of Morphological Classifications for nearly 900,000 galaxies,” 2010.