On the Cost Complexity of Crowdsourcing

Yili Fang, Hailong Sun*, Pengpeng Chen, Jinpeng Huai
SKLSDE, School of Computer Science and Engineering, Beihang University, Beijing, China
Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing, China
{fangyili,chenpp}@act.buaa.edu.cn, {sunhl,huaijp}@buaa.edu.cn

Abstract

Existing efforts mainly use empirical analysis to evaluate the effectiveness of crowdsourcing methods, which is often unreliable across experimental settings. Consequently, it is of great importance to study theoretical methods. This work, for the first time, defines the cost complexity of crowdsourcing, and presents two theorems to compute the cost complexity. Our theorems provide a general theoretical method to model the trade-off between costs and quality, which can be used to evaluate and design crowdsourcing algorithms, and characterize the complexity of crowdsourcing problems. Moreover, following our theorems, we prove a set of corollaries that can obtain existing theoretical results for special cases. We have verified our work theoretically and empirically.

1 Introduction

Crowdsourcing provides an effective means for solving many real-world problems, e.g. labeling training data for machine learning. As crowdsourcing workers are normally non-experts, the individual contributions from one worker can be unreliable. In practice, task redundancy is commonly used to amortize the unreliability in crowdsourcing with extra costs.

Many efforts [Ho *et al.*, 2013; Roy *et al.*, 2015; Yu *et al.*, 2017; Tran-Thanh *et al.*, 2013] have been made to obtain high quality results with as few costs as possible, where the quality is often measured with the result error rate and the costs are evaluated with the times of querying humans. To evaluate the effectiveness of the proposed methods, prior works mainly employ experimental analysis. However, the same methods often exhibit contradicting results in different experiments [Liu *et al.*, 2012; Zhou *et al.*, 2015; Li and Liu, 2015]. For instance, [Zhou *et al.*, 2012] shows that the Dawid & Skene (DS) model [Dawid and Skene, 1979] outperforms majority voting (MV) in terms of result accuracy while [Han *et al.*, 2016] presents exactly reverse results in tasks for acquiring specific knowledge. [Liu *et al.*, 2012] demonstrates that the homogeneous DS (HDS) model [Raykar *et al.*, 2010] is better than MV in *bluebird* dataset while MV is better than HDS in *rte* and *temp*. Therefore, empirical analysis is not a reliable means to evaluate crowdsourcing methods for general cases, which calls for theoretical research to overcome such limitations. In this regard, the NSF computing division [Wing, 2008] considers the development of theoretical tools for systems involving human computation as one of the five major challenges that today's computing faces.

Inspired by the theorectical computer science, some efforts [Shahaf and Amir, 2007; Kulkarni, 2011] concern the theoretical models involving humans in computation. [Shahaf and Amir, 2007] presents a Human-Assisted Turing Machine (HTM) that models the hybrid computation paradigm with machines and humans. Basing on HTM, the authors discuss how to measure human efforts such as the times and size of human input so as to define the algorithm and problem complexity, but they do not consider the unreliability of workers or the trade-off of costs and quality in crowdsourcing. Furthermore, [Kulkarni, 2011] discusses the importance of building a theoretical model of computation involving humans in terms of algorithm evaluation, algorithm comparison and cost quantification. Another line of theoretical research efforts [Li and Liu, 2015; Wang and Zhou, 2016; Gao and Zhou, 2016; Gao *et al.*, 2016] focus on estimating the upper bound on the mean error rate of specific algorithms. [Li and Liu, 2015] gives the upper bound on the mean error rate with weighted majority voting (WMV). [Wang and Zhou, 2016] theoretically analyzes the costs and the result error rate of MV. And [Gao and Zhou, 2016; Gao *et al.*, 2016] theoretically studies the performance of the DS model, but ignores the parameter learning that can greatly affects the result accuracy. [Nushi *et al.*, 2015; Venanzi *et al.*, 2014] addresses the parameter learning issue in the context of data sparsity, which is helpful for better algorithm designing. In summary, although existing efforts recognize the importance of theoretical models in human participant computation systems like crowdsourcing, few provide general theoretical methods for crowdsourcing.

In this work, motivated by the classical computational complexity, the sample complexity and the PAC theory in machine learning [Balcan *et al.*, 2010], we study a general theoretical approach to understanding the complexity of crowdsourcing that is affected by multiple interplaying factors such as number of workers, worker ability and aggregation methods. Specifically, we propose the cost complexity of crowd-

*Corresponding author

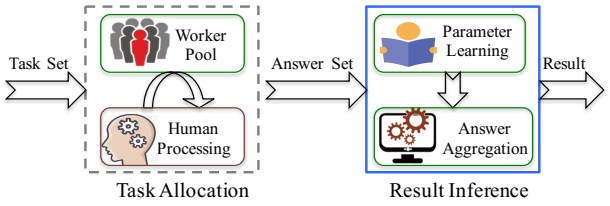

Figure 1: The framework of a crowdsourcing workflow.

sourcing to theoretically address the trade-off of costs and quality in crowdsourcing. We argue that like computational complexity, the cost complexity of crowdsourcing is useful for evaluating, comparing and designing of algorithms, measuring the complexity of crowdsourcing problems and etc.

Our major contributions are as follows:

- We define the cost complexity of crowdsourcing, which measures how many costs are needed to meet certain quality requirements. To the best of our knowledge, this is the first effort to give a formal definition of the crowdsourcing complexity that relates costs with quality.
- We give two theorems that can derive the cost complexity of general crowdsourcing. For a specific crowdsourcing algorithm, our method not only can give the theoretical upper bound on the mean error rate, but can estimate how many workers to hire for achieving a certain quality objective. Following the theorems, we also obtain a set of corollaries that have been verified in existing work.
- Through a set of case studies, we have verified our method through theoretical analysis and experimental evaluation on real-world datasets. The outcome explains the contradicting results in previous work. For instance, when each worker completes more than $\frac{1}{(\eta^{\text{MV}}-\eta^{\text{HDS}})^2}\ln\frac{2|\mathcal{H}|}{\delta}$) tasks, HDS will outperform MV; otherwise, MV is better.

2 Overview of Crowdsourcing Workflows

Fig.1 shows the general framework of crowdsourcing workflows consisting of *task allocation* and *result inference*.

2.1 Task Allocation

Task allocation involves worker pool construction and task assignment. First, a high-quality worker pool can be built by filtering out low-ability workers with qualification tests [AMT, 2017; Ipeirotis and Gabrilovich, 2014; Marcus *et al.*, 2015] or by analyzing historical logs [Jung, 2014; Ambati *et al.*, 2011]. Then the distribution of the workers in worker pool over ability can be obtained through qualification tests or log analysis by statistical methods. For instance, Quizz [Ipeirotis and Gabrilovich, 2014] obtains the Beta distribution of worker ability with qualification tests. Next, in task assignment, a task is assigned to a certain number of workers selected from the work pool with a specific strategy. Then with the constructed worker pool and the adopted task assignment strategy, we can obtain the distribution of participating workers over ability (denoted by $\mathcal{W}$) in task processing. And for result inference, $\mathcal{W}$ can be considered the prior knowledge.

Assume that there are m workers and n tasks. We denote the ground truth set by $\mathcal{Y} = \{y_j | 0 < j < n\}$, and the truth answer to task j by y_j that takes on a value in a candidate answer set $\mathcal{A} = \{0, \cdots, K-1\}$. Let $X = \{x_{ij} | i \leq m, j \leq n\}$ be the answer set of n tasks from m workers, x_{ij} be the answer given by worker i to task j, and X_j denote the answer set of task j. We use the confusion matrix in (1) to characterize workers' abilities.

$$\pi_{kl}^{ij} = \mathbb{P}(x_{ij} = l | y_j = k), \tag{1}$$

which satisfies $\sum_{l=1}^{L} \pi_{kl}^{ij} = 1$. Given $y_j = k$, x_{ij} is generated by a multinomial distribution with $\pi_{k*}^{ij} = (\pi_{k1}^{ij}, \cdots, \pi_{kL}^{ij})$. Thus the three-dimensional matrix $\pi^{(i)} = [\pi_{kl}^{ij}]$ denotes the abilities of worker i in all tasks. After task allocation, we may not know π, but we can know its distribution $\mathcal{W}$.

2.2 Result Inference

Result inference is to infer the truth result of a task by aggregating the answers. There are mainly two lines of work. 1) *Voting.* Majority voting, as the simplest voting method, infers the final result by simply counting the votes for each alternative answer [Snow *et al.*, 2008; Ipeirotis and Gabrilovich, 2014]. Though simple, it suffers from being error-prone due to the ignorance of the difference of workers' abilities and other parameters. In light of this, weighted majority voting [Li and Liu, 2015] incorporates workers' abilities into majority voting. Specifically, it assigns different weights to votes according to workers' abilities that are unknown parameters. 2) *Probabilistic approach.* Probabilistic generative models containing unknown parameters (e.g. worker ability) are employed to specify workers' performance on tasks, and then the parameters are estimated (parameter learning), finally the answers are aggregated through model inference, e.g. inferring the final result by using EM algorithm for parameter estimation of probabilistic generative models [Dawid and Skene, 1979; Raykar *et al.*, 2010; Salek *et al.*, 2013].

Without loss of generality, we define a unified aggregation function $f : \mathcal{A}^{|X_j|} \to \mathcal{A}$ as follows:

$$f(X_j) = \underset{k \in \mathcal{A}}{\text{argmax}} \sum_{i=1}^{m} As_j(i, k, x_{ij}), \tag{2}$$

where $As_j(i, k, x_{ij})$ is the aggregation score when worker i gives answer $x_{ij} \in \mathcal{A}$ to task j the ground truth of which is $k \in \mathcal{A}$. (2) is a universal representation of result inference. For majority voting, we have $As_j(i, k, x_{ij}) = \mathbf{I}(x_{ij} = k)$, where $\mathbf{I}(\cdot)$ is an indicator function; and for weighted majority voting, $As_j(i, k, x_{ij}) = v_i \mathbf{I}(x_{ij} = k)$, where v_i is the weight of worker i which can be obtained with machine learning. In probabilistic methods (e.g. DS model), $As_j(i, k, l) = \log \pi_{kl}^{ij}$ (we use l to mark x_{ij}), where $\pi_{kl}^{ij} \in \pi^{(i)}$ denotes the worker ability that needs to be estimated with machine learning.

Let $\mathcal{D}$ denote the distribution of the ground truth among the candidate answers. We define the loss function to measure the mean error rate of f as follows:

$$L_{(\mathcal{D},\mathcal{W},\mathcal{Y})}(f) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{P}\{f(X_j) \neq y_j\}. \tag{3}$$

However, to evaluate the effectiveness of f, we first need to learn the unknown parameters in f. Straightforwardly, the more workers' answers we have, the better we can learn the parameters of f, which is exactly what sample complexity addresses. In computational learning theory, probably approximately correct (PAC) learning is a framework for mathematical analysis of the sample complexity of machine learning algorithms. $\mathcal{H}$ is a set of answer aggregation functions with different parameters, the input and output of which are X and $\mathcal{Y}$ respectively. $\mathcal{H}$ is called the hypothesis class and every member in $\mathcal{H}$ is called a hypothesis. f is an unknown answer aggregation function $f : \mathcal{A}^{|X_j|} \to \mathcal{A}$. For simplicity, it is assumed that $f \in \mathcal{H}$, which is called the realizability assumption. A learner is given access to an oracle $EX(\mathcal{D}, \mathcal{W}, f)$, which outputs aggregated answers one at a time randomly and independently according to $\mathcal{D}$, $\mathcal{W}$ and f. The goal is to learn f from $\mathcal{H}$ so that the corresponding parameters can be correctly learned.

$S = (X_1, X_2, \cdots X_n)$ is a finite sequence of answers set for all tasks. This is the learner's input and is generated by n calls to $EX(\mathcal{D}, \mathcal{W}, f)$. The learner's output is $h \in \mathcal{H}$, h is the answer aggregation function with the estimated value of parameters. To measure the effectiveness of the learner, we define the error of a function h as follows:

$$L_{(\mathcal{D},\mathcal{W},f)}(h) = \mathbb{P}_{(\mathcal{D},\mathcal{W})}\{h(X_j) \neq f(X_j)\}. \tag{4}$$

(4) denotes the probability that h disagrees with f on distribution $\mathcal{D}$ and $\mathcal{W}$. Ideally, h agrees with f in the whole domain, namely, $L_{(\mathcal{D},\mathcal{W},f)}(h) = 0$. That is to say the unknown parameters are accurately learned, and the final error rate only depends on answer aggregation function. For instance, majority voting can be viewed as such a special case, where all parameters are known in f.

3 The Cost Complexity of Crowdsourcing

We use the terminology, *cost complexity*, to denote the costs of solving a crowdsourcing task. Specifically, it measures the number of task requests for human workers. Different from the traditional computational complexity, the cost complexity of crowdsourcing is closely related to the quality requirement and workers' abilities. This section introduces two formal definitions of the cost complexity of crowdsourcing given a certain quality constraint on the mean error rate and the distribution over workers' abilities.

Let U denote the set of workers selected by task allocation.

Definition 1. *If there is a learning algorithm $A(m', n', \delta)$ that outputs an aggregation function h and three values $\eta_U \in \mathcal{R}$, $\epsilon_{n'} \in \mathcal{R}$, $\eta_{m'} \in \mathcal{R}$ after making at most c ($c = m' \times n'$) task requests, such that for any answer aggregation function $f \in \mathcal{H}$, $\epsilon \in (0, 1/2)$, $\eta \in (0, 1/2)$, $\delta \in (0, 1/4)$, for any $m' \geq 0$, any worker set U, we have $\mathbb{P}\{L_{(\mathcal{D},\mathcal{W},f)}(h) \leq \epsilon_{n'}\} \geq 1 - \delta$ and $L_{(\mathcal{D},\mathcal{W},\mathcal{Y})}(f) \leq \eta_U$; and for any $\eta_{m'} = \exp(\mathbb{E}_{\mathcal{W}}(\ln(\eta_U)))$ and $c \geq \mathcal{O}^{\mathcal{W}}(\epsilon, \delta, f, \eta)$, we have*

$$\begin{cases} \mathbb{P}\{L_{(\mathcal{D},\mathcal{W},f)}(h) \leq \epsilon_{n'} \leq \epsilon\} \geq 1 - \delta, \\ L_{(\mathcal{D},\mathcal{W},\mathcal{Y})}(f) \leq \eta_U \text{ and } \eta_{m'} \leq \eta. \end{cases} \tag{5}$$

Then we call $\mathcal{O}^{\mathcal{W}}(\epsilon, \eta, \delta, f)$ the cost complexity of crowdsourcing with worker distribution $\mathcal{W}$ over abilities.

Essentially $\mathcal{O}^{\mathcal{W}}$ measures how many answers should be solicited from workers for learning an aggregation function and infering the final crowdsourcing results accurately. $\epsilon_{n'}$ and $\eta_{m'}$ respectively specify the error rate bounds of the parameter learning of the aggregation function f and the aggregated results. η_U is the error rate bound of the aggregated results determined by the worker set U and is related to $\eta_{m'}$. Actually $\mathcal{O}^{\mathcal{W}}$ borrows the concept of sample complexity from machine learning. $\mathcal{O}^{\mathcal{W}}$ depends on m', the total number of workers, and n', the number of tasks a worker completes.

A special case of Definition 1 is when all workers exhibit the same ability w, which is not unusual because many microtasks do not require much expertise and all workers can fulfill the tasks. For this case, we define $\mathcal{O}^{w}$ as follows:

Definition 2. *If there is a learning algorithm $A(m', n', \delta)$ that outputs an aggregation function h and two values $\eta_{m'} \in \mathcal{R}$, $\epsilon_{n'} \in \mathcal{R}$ after making at most c ($c = m' \times n'$) task requests, such that for any answer aggregation function $f \in \mathcal{H}$, $\epsilon \in (0, 1/2)$, $\eta \in (0, 1/2)$, $\delta \in (0, 1/4)$, for any $m' \geq 0$, $n' \geq 0$, we have $\mathbb{P}\{L_{(\mathcal{D},w,f)}(h) \leq \epsilon_{n'}\} \geq 1 - \delta$, and $L_{(\mathcal{D},w,\mathcal{Y})}(f) \leq \eta_{m'}$; and for any $c \geq \mathcal{O}^{w}(\epsilon, \delta, f, \eta)$, we have*

$$\begin{cases} \mathbb{P}\{L_{(\mathcal{D},w,f)}(h) \leq \epsilon_{n'} \leq \epsilon\} \geq 1 - \delta, \\ L_{(\mathcal{D},w,\mathcal{Y})}(f) \leq \eta_{m'} \leq \eta. \end{cases} \tag{6}$$

Then we call $\mathcal{O}^{w}(\epsilon, \eta, \delta, f)$ the cost complexity of crowdsourcing with identical workers' ability w.

From Definition 2, we know that $\eta_{m'}$ and $\epsilon_{n'}$ only depends on m' and n' respectively. A simple scenario is when f has no unknown parameters (e.g., MV), which means $L_{(\mathcal{D},w,f)}(h) = 0$ always hold. In that case, c only depends on m', and $\mathcal{O}^{\mathcal{W}}$ only depends on $\eta_{m'}$.

Note that our cost complexity specifies the max number of workers needed to achieve an error rate bound. In other word, c is at least $\mathcal{O}^{\mathcal{W}}$, but we do not require workers complete the same number of tasks. n' can be regarded as the min number of tasks a worker must complete. In reality, if a worker completes over n' tasks, the practical cost is less than that given by the cost complexity for a quality constraint.

4 Main Results

The *cost complexity* is closely related to the quality requirements measured by error rate $\eta_{m'}$ and $\epsilon_{n'}$. This section first presents Theorem 1 and Theorem 2 to compute $\mathcal{O}^{\mathcal{W}}$ and the upper bound on the error rate respectively, which are the main contributions of this work.

Theorem 1. *Given a task j ($1 \leq j \leq n$) processed by m workers, we can obtain an answer set $X_j = \{x_{ij} | i \leq m\}$. For simplicity, we use l to mark x_{ij}. Worker i has ability $\pi^{ij} = [\pi^{ij}_{kl}]$ when processing task j, π^{ij}_{kl} is the performance when worker i processes task j the truth answer of which is $k \in \mathcal{A}$. Let $f(X_j)$ denote the aggregation method. Then the complexity $\mathcal{O}^{\mathcal{W}}$ can be computed as follows:*

$$\mathcal{O}^{\mathcal{W}}(\epsilon, \eta, \delta, f) = -\frac{2}{\epsilon^2(1-2\eta)^2}\ln(\frac{2|\mathcal{H}|}{\delta}) \times \frac{2(\ln\frac{\eta}{K-1})(a-b)^2 - \mathbb{D}_{\mathcal{W}}(\mu_i)}{\mathbb{E}^2_{\mathcal{W}}(\mu_i)}, \tag{7}$$

where $\mu_i = \sum_l \pi_{kl}^{ij}(As_j(i,g,l) - As_j(i,k,l))$ *is a value specific to worker* i, $As_j(i,g,l) \in [a,b]$, *and* g *represents any alternative answer other than the truth (i.e.* $g \neq k$*), and* $\mathbb{D}_{\mathcal{W}}(\mu_i)$ *and* $\mathbb{E}_{\mathcal{W}}(\mu_i)$ *are the variance and expectation of* μ_i *respectively.*

Remarks. This theorem provides a theoretical means to estimate how many queries of human workers should be made for meeting certain quality requirements. Its benefits are three-fold. First, given a set of workers with distribution $\mathcal{W}$ of workers over ability, Theorem 1 can be used to compare the performance of two result aggregation functions. Second, given a result aggregation function, Theorem 1 helps evaluate the performance of different task allocation methods. Third, when both task allocation and result aggregation methods are given, Theorem 1 can estimate how many workers should be hired. In all, Theorem 1 provides a fundamental tool for designing and evaluating general crowdsourcing solutions.

$\mathcal{O}^{\mathcal{W}}$ depends on the error rate bounds (η_U and $\epsilon_{n'}$) of result inference and the parameter learning of the aggregation function. To prove Theorem 1, we first give Theorem 2 that analyzes the error rate bound of result inference.

Theorem 2. *For the crowdsourcing tasks described in Theorem 1, the error rate of the aggregated results can be upper-bounded as follows:*

$$L_{(\mathcal{D},\mathcal{W},\mathcal{Y})}(f) \leq (K-1)\exp\left(-\frac{(\sum_{i=1}^{m}\mu_i)^2}{2m(a-b)^2}\right), \quad (8)$$

where K is the size of the answer set $\mathcal{A}$.

Proof. First, we give a general function to denote an aggregation process.

$$f(X_j) = \underset{k\in\mathcal{A}}{\operatorname{argmax}} \sum_{i=1}^{m} As_j(i,k,x_{ij}).$$

Let $As_j(i,k,l) \in [a,b]$, $Z_i^{gk} = As_j(i,g,l) - As_j(i,k,l)$, then $Z_i^{gk} \in [a-b, b-a]$ and the expectation is $\mu_i = \sum_l \pi_{kl}^{ij}(As_j(i,g,l) - As_j(i,k,l))$. First we apply the union bound to get (9) and obtain (10) with Hoeffding's inequality. We get the error rate bound as follows:

$$\begin{aligned}
&L_{(\mathcal{D},\mathcal{W},\mathcal{Y})}(f) = \mathbb{P}\{g \neq k, \sum_i^m Z_i^{gk} \geq 0\} \\
&\leq \sum_{g\in\mathcal{A}}^{g\neq k} \mathbb{P}\{\sum_i^m Z_i^{gk} \geq 0\} \quad (9) \\
&\leq \sum_{g\in\mathcal{A}}^{g\neq k} \exp\left(-\frac{(\mathbb{E}_l \sum_{i=1}^{m}(As_j(i,g,l) - As_j(i,k,l)))^2}{2\sum_{i=1}^{m}(a-b)^2}\right) \quad (10) \\
&= \sum_{g\in\mathcal{A}}^{g\neq k} \exp\left(-\frac{(\sum_{i=1}^{m}\sum_l \pi_{kl}^{ij}(As_j(i,g,l) - As_j(i,k,l)))^2}{2m(a-b)^2}\right) \\
&\leq (K-1)\exp\left(-\frac{(\sum_{i=1}^{m}\mu_i)^2}{2m(a-b)^2}\right).
\end{aligned}$$

Thus, $L_{(\mathcal{D},\mathcal{W},\mathcal{Y})}(f) \leq (K-1)\exp\left(-\frac{\left(\sum_{i=1}^{m}\mu_i\right)^2}{2m(a-b)^2}\right)$. □

Theorem 2 provides a general method to compute the upper bounds of error rates η_U in an aggregation function with the worker set U. Except for result inference, task allocation determines worker distribution $\mathcal{W}$ and can greatly affect the crowdsourcing results, and this is what Theorem 1 addresses on the basis of Theorem 2.

For identical worker abilities, we can generalize Theorem 2 to obtain Corollary 1 given in [Wang and Zhou, 2016].

Corollary 1. *Given* m' *workers whose abilities are i.i.d. according to parameters* $q = [q_0, q_1, \cdots, q_{K-1}]$, *the ground-truth label* $i^* \in \{0, 1, \cdots, K-1\}$ *and* $\gamma = min_{i\neq i^*}(q_{i^*} - q_i) > 0$. *For the error rate of the aggregated results to be upper-bounded by* η, *it is sufficient that*

$$m' \geq \frac{2}{\gamma^2}\ln\left(\frac{K-1}{\eta}\right). \quad (11)$$

We can further generalize Theorem 2 by using the weighted majority voting (WMV) with weight $K\hat{w}_i - 1$ as the aggregation function. Then we can obtain Corollary 2 that is given in [Li and Liu, 2015].

Corollary 2. *For a set of* m *workers* U, *using the weighted majority voting with weights* $K\hat{w}_i - 1$, *aggregation result* $\hat{y}_j$ *of task* j. *And an unbiased estimator of the workers' ability* $\mathbb{E}(\hat{w}_i) = w_i$ *that satisfies* $\{w_i\}_{i\leq K-1}$. *If the workers' labels are generated independently according to the following probability:*

$$\pi^{ij} = \begin{bmatrix} w_i & \frac{1-w_i}{K-1} & \cdots & \frac{1-w_i}{K-1} \\ \frac{1-w_i}{K-1} & w_i & \cdots & \frac{1-w_i}{K-1} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1-w_i}{K-1} & \frac{1-w_i}{K-1} & \cdots & w_i \end{bmatrix}. \quad (12)$$

Then we have

$$\frac{1}{n}\sum_{j=1}^{m}\mathbb{P}\{\hat{y}_j \neq y_j\} \leq exp\left(-\frac{2F(U)^2}{K^2(K-1)^2} + \ln(K-1)\right),$$

where $F(U) = \frac{1}{\sqrt{m}\sum_{i\leq m}(Lw_i-1)^2}$.

With Corollary 2, we can learn that weighted majority voting can achieve the same error rate as majority voting if the weight (worker ability) is a constant value.

However, if workers' abilities are unknown, analyzing the weighted majority voting entails considering the parameter learning process. In practice, the parameter learning is indispensable for many aggregation methods. And it is difficult to learn the unknown parameters 100% accurately due to the limitation of dataset and machine learning algorithms.

Regarding Definition 1, $\mathcal{O}^{\mathcal{W}}$ involves parameter learning which also affects the error rates of aggregation method with estimated value of unknown parameters. Thus, we use PAC learnability to analyze the mean error rate of aggregation methods. First, we formulate the error rate of inferred results by considering parameter learning.

$$\begin{aligned}
err(h) &= L_{(f,\mathcal{Y})}(1 - L_{(h,f)}) + (1 - L_{(f,\mathcal{Y})})L_{(h,f)} \\
&= (1 - 2L_{(h,f)})L_{(f,\mathcal{Y})} + L_{(h,f)}, \quad (13)
\end{aligned}$$

where $L_{(f,\mathcal{Y})}$ and $L_{(h,f)}$ are the abbreviations of $L_{(\mathcal{D},\mathcal{W},\mathcal{Y})}(f)$ and $L_{(\mathcal{D},\mathcal{W},f)}(h)$ respectively. In some special cases, some aggregation methods, such as majority voting, do not entail parameter learning due to the absence of parameters in their aggregation rules. In this case, $err(h)$ is identical to $L_{(f,\mathcal{Y})}$.

To prove Theorem 1, we still need to introduce the Valiant's PAC (Lemma 1) to compute n' for achieving the error rate $\epsilon_{n'}$ of parameter learning.

Lemma 1. *[Angluin and Laird, 1988] Let $\mathcal{H}$ be a finite hypothesic class, let $\delta, \epsilon \in (0,1), \eta \in (0, \frac{1}{2})$ and let n' be an integer that satisfies*

$$n' \geq \frac{2}{\epsilon^2(1-2\eta)^2} \ln(\frac{2|\mathcal{H}|}{\delta}). \tag{14}$$

Then, for any f and $\mathcal{D}$, for which the realizability assumption holds, given a sequence S of size n', if a hypothesis $h \in \mathcal{H}$ minimizes $err(h)$, we have

$$\mathbb{P}\{L_{(\mathcal{D},\mathcal{W},f)}(h) \geq \epsilon\} \leq \delta. \tag{15}$$

Although the parameter space can be infinite, the hypothesis space $\mathcal{H}$ is finite. For instance, given n tasks, m workers and answer set $A(|A| = K)$, a hypothesis $h \in \mathcal{H}$ is actually a mapping from workers' answer set to a set of truth answers. Thus the size of the hypothesis space is $|K|^n$ at most. Then we can use VC theory [McAllester, 1998] to obtain $|\mathcal{H}|$.

As for crowdsourcing tasks, the result aggregation method can be viewed as a process of machine Learning. The error rate is η, if the parameters are accurately learned. In particular, $err(h)$ is affected by the error rate of aggregation rule η (i.e. $err(h) = L_{(f,\mathcal{Y})}$) when the parameter learning process generates no error, i.e. err ($L_{h,f)} = 0$). Most existing literatures analyze the result aggregation rule with the impractical assumption that $L_{(\mathcal{D},\mathcal{W},f)}(h) = 0$.

When each worker has the identical ability (i.e., $w_1 = w_2, \cdots, w_m$), for task j, more than $\frac{(b-a)^2}{2\mu_i^2} \ln \frac{|L|-1}{\eta}$ workers can generate the error rate less than η with the respect to the aggregation rule, where $\mu_i = \sum_{l=1}^{L} \pi_{kl}^{ij}(As_j(i,g,l) - As_j(i,k,l))$.

Next we employ PAC learnability to prove Theorem 1.

Proof of Theorem 1. Basing on Theorem 2, we set $\mu_i = \sum_l \pi_{kl}^{ij}(As_j(i,g,l) - As_j(i,k,l))$. Workers' abilities vary with different workers in Definition 1. We can obtain

$$L_{(f,\mathcal{Y})} \leq (K-1)\exp(-\frac{(\sum_{i=1}^{m'} \mu_i)^2}{2m'(a-b)^2}) = \eta_U.$$

Since $\eta_{m'} = \exp(\mathbb{E}_{\mathcal{W}}(\ln(\eta_U))$ and $\eta_{m'} \leq \eta$ in Definition 2, we can obtain

$$\mathbb{E}_{\mathcal{W}}(\sum_{i=1}^{m} \mu_i)^2 \geq -2m'(\ln \frac{\eta}{K-1})(a-b)^2.$$

Based on the property of variance $\mathbb{E}_{\mathcal{W}}(\sum_{i=1}^{m} \mu_i)^2 - \mathbb{E}^2_{\mathcal{W}}(\sum_{i=1}^{m} \mu_i) = \mathbb{D}_{\mathcal{W}}(\sum_{i=1}^{m} \mu_i)$, we get

$$\mathbb{E}^2_{\mathcal{W}}(\sum_{i=1}^{m'} \mu_i) + \mathbb{D}_{\mathcal{W}}(\sum_{i=1}^{m'} \mu_i) \geq -2m(\ln \frac{\eta}{K-1})(a-b)^2.$$

Since all workers come from the same worker pool following a certain distribution $\mathcal{W}$ over ability. Meanwhile, all $\mathbb{E}_{\mathcal{W}}(\mu_i)$s are equal, all $\mathbb{D}_{\mathcal{W}}(\mu_i)$s are fixed, and workers are independent of each other. Then we can get

$$m'^2\mathbb{E}^2_{\mathcal{W}}(\mu_i) + m'\mathbb{D}_{\mathcal{W}}(\mu_i) \geq -2m'(\ln \frac{\eta}{K-1})(a-b)^2.$$

It can be simplified as

$$m' \geq -\frac{2(\ln \frac{\eta}{K-1})(a-b)^2 - \mathbb{D}_{\mathcal{W}}(\mu_i)}{\mathbb{E}^2_{\mathcal{W}}(\mu_i)}.$$

Since the aggregation function contains parameters, we need parameter learning. Then based on Lemma 1, we get

$$n' \geq \frac{2}{\epsilon^2(1-2\eta_b)^2} \ln(\frac{2|\mathcal{H}|}{\delta}).$$

As $c = m'n'$ in Definition 1, we can derive $\mathcal{O}^{\mathcal{W}} - \frac{2}{\epsilon^2(1-2\eta)^2} \ln(\frac{2|\mathcal{H}|}{\delta}) \frac{2(\ln \frac{\eta}{K-1})(a-b)^2 - \mathbb{D}_{\mathcal{W}}(\mu_i)}{\mathbb{E}^2_{\mathcal{W}}(\mu_i)}$. □

Corollary 3. *Let $f(X_j)$ be the aggregation function of crowdsourcing for each task j ($As_j(i,g,l) \in [a,b]$), suppose all workers have identical ability π, then the cost complexity of crowdsourcing $\mathcal{O}^w$ can be computed as follows:*

$$\begin{aligned}\mathcal{O}^w(\epsilon,\eta,\delta,f) =& \frac{2}{\epsilon^2(1-2\eta)^2} \ln(\frac{2|\mathcal{H}|}{\delta}) \\ & \times \frac{(b-a)^2}{2w^2} \ln \frac{K-1}{\eta},\end{aligned} \tag{16}$$

where $w = \sum_l \pi_{kl}^{ij}(As_j(i,g,l) - As_j(i,k,l))$ is the same for all workers.

This section mainly gives a general method to compute the cost complexity of crowdsourcing, which is determined by the distribution of worker over ability, the parameter learning and answer aggregation in result inference. Theorem 1 and 2, the major contributions of this work, provide a method to compute the cost complexity and the upper bound on the mean error rate respectively for general crowdsourcing workflows. Corollary 1-3 give some interesting results that are obtained by generalizing the two theorems to special cases.

Due to space limitation, the proofs of Corollary 1-3 are shared on the web [1].

5 Case Studies

In this section, we aim at verifying the effectiveness of Theorem 1 through applying it to three representative result aggregation algorithms, including MV, WMV and HDS. Specifically, we conducted case studies both theoretically and empirically.

5.1 Theoretical Analysis

First, for MV, all parameters with f are known. In other words, h is equal to f. Then, we have Corollary 4.

[1] Link to the proofs of Corollary 1-3: https://goo.gl/ZhKddo

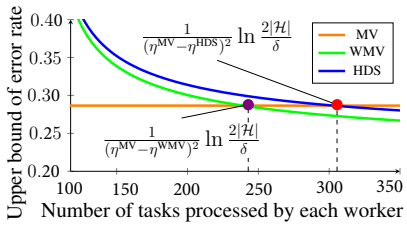

Figure 2: The upper bound of error rate for three methods.

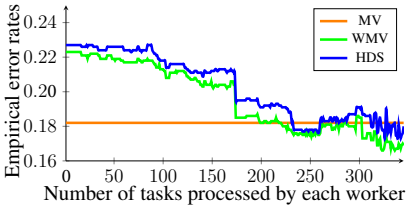

Figure 3: The error rate analysis on *dog* dataset.

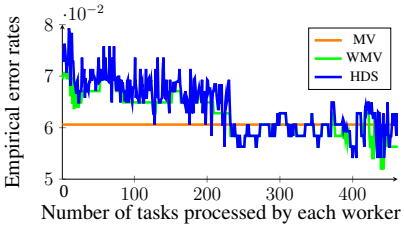

Figure 4: The error rate analysis on *temp* dataset.

Corollary 4. *Let* $\mu_i = 1 - 2w_i$*, then* $\mathcal{O}^{\mathcal{W}}$ *for MV* $f = \text{argmax}_k \sum_{i=1}^{m} \mathbf{I}(x_{ij} = k)$ *is computed as follows:*

$$\mathcal{O}^{\mathcal{W}}(\epsilon,\eta,\delta,f) = -\frac{(2\ln\frac{\eta}{K-1}) - \mathbb{D}_{\mathcal{W}}(\mu_i)}{\mathbb{E}^2_{\mathcal{W}}(\mu_i)}. \quad (17)$$

Similarly, we can obtain Corollary 5 for WMV:

Corollary 5. *Let* $\mu_i = w_i(1-2w_i)$*, then* $\mathcal{O}^{\mathcal{W}}$ *for WMV* ($f = \text{argmax}_k \sum_{i=1}^{m} w_i \mathbf{I}(x_{ij} = k)$) *is as follows:*

$$\mathcal{O}^{\mathcal{W}}(\epsilon,\eta,\delta,f) = -\frac{2}{\epsilon^2(1-2\eta)^2}\ln(\frac{2|\mathcal{H}|}{\delta}) \times \frac{2(\ln\frac{\eta}{K-1}) - \mathbb{D}_{\mathcal{W}}(\mu_i)}{\mathbb{E}^2_{\mathcal{W}}(\mu_i)}. \quad (18)$$

Furthermore, for HDS, we have Corollary 6:

Corollary 6. *Let* μ_i *mark* $\ln\frac{w_i}{1-w_i}(1-2w_i)$*,* $f(X_j)$ *be the aggregation function of HDS, then* $\mathcal{O}^{\mathcal{W}}$ *is as below:*

$$\mathcal{O}^{\mathcal{W}}(\epsilon,\eta,\delta,f) = -\frac{2}{\epsilon^2(1-2\eta)^2}\ln(\frac{2|\mathcal{H}|}{\delta}) \times \frac{2(\ln\frac{\eta}{K-1}) - \mathbb{D}_{\mathcal{W}}(\mu_i)}{\mathbb{E}^2_{\mathcal{W}}(\mu_i)}. \quad (19)$$

In Equation (13), there involve two types of error rate including parameter learning and answer aggregation. Suppose we fix the number of workers processing one task, then we can obtain the upper bound of error rate in answer aggregation function. As shown in Fig.2, we plot the upper bound on the mean error rate obtained theoretically from the three algorithms including MV, WMV and HDS. We can observe that the upper bound on the mean error rate in MV is stable (i.e. the error rate upper bound is η^{MV}) varying with the number of tasks processed by each worker. While the the error rate upper bound with WMV/HDS decreases as number of tasks grows, and WMV outperforms HDS. When the number of tasks processed by each worker is less than $\frac{1}{(\eta^{\text{MV}}-\eta^{\text{WMV}})^2}\ln\frac{2|\mathcal{H}|}{\delta}$ ($\frac{1}{(\eta^{\text{MV}}-\eta^{\text{HDS}})^2}\ln\frac{2|\mathcal{H}|}{\delta}$), MV outperforms WMV/HDS, which well explains the contradicting results in existing work [Zhou *et al.*, 2012] and [Han *et al.*, 2016].

Note: as Corollary 4-6 can be proved by simply substituting the expectation and variance of the worker distribution over ability (i.e., (12)) into Theorem 1, we omit the proofs.

5.2 Emprical Analysis

Here we present the experimental analysis of error rates with two real-world crowdsourcing datasets: *dog* [Zhou *et al.*, 2012] and *temp* [Snow *et al.*, 2008]. The maximum number of tasks processed by a worker are 345 and 462 respectively for the two datasets. For each dataset, we first grouped the answers into subsets according to worker IDs, then varied the number of tasks that a worker processes. Next we inferred the results with three algorithms including MV, WMV and HDS, and we computed the corresponding error rate. The results are plotted in Fig.3 and Fig.4. As the results presented here are about error rate instead of the upper bound on the error rate shown in Fig.2, the plots fluctuate with the increase of task amount, but they demonstrate similar trends to our theoretical analysis shown in Fig.2.

Note in our experiments, for a specific number of tasks x, if a worker finishes over x' ($x' > x$) tasks, we will replace that worker with a group of simulated workers. Each simulated worker finishes x or ($x' \mod x$) tasks and the total number of tasks they finish is exactly x'. For instance, for $x = 10$, if a worker finishes 23 tasks in reality, three workers who finishe 10, 10 and 3 tasks respectively will be generated.

6 Conclusion

This work studies the computational complexity of general crowdsourcing workflows. We first give two definitions of the cost complexity of crowdsourcing, i.e. $\mathcal{O}^{\mathcal{W}}$ and $\mathcal{O}^{w}$, for different distribution over workers' abilities. Then we present two theorems to compute the cost complexity and the upper bound on the mean error rate respectively for general crowdsourcing workflows. We further generalize our theoretical results to special cases and obtain a set of corollaries that have been verified in existing work. Finally, to verify the effectiveness of our methods, we present a set of case studies for three representative answer aggregation methods both theoretically and empirically, which explains the existing contradicting experimental results. In all, our work benefits crowdsourcing in designing and evaluating crowdsourcing algorithms.

Acknowledgements

This work was supported partly by National Basic Research Program of China (973 Program) under Grant Nos. 2015CB358700 and 2014CB340304, partly by National Key Research and Development Program of China under Grant No.2016YFB1000804, and partly by National Natural Science Foundation under Grant No. 61421003.

References

[Ambati *et al.*, 2011] Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. Towards task recommendation in micro-task. In *HCOMP*, pages 80–83, 2011.

[AMT, 2017] AMT developer guide. http://docs.aws.amazon.com/AWSMechTurk/latest/AWSMechanicalTurkRequester/amt-dg.pdf, 2017.

[Angluin and Laird, 1988] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[Balcan *et al.*, 2010] Maria Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2):111–139, 2010.

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

[Gao and Zhou, 2016] Chao Gao and Dengyong Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. *Statistics*, 2016.

[Gao *et al.*, 2016] Chao Gao, Yu Lu, and Dengyong Zhou. Exact exponent in optimal rates for crowdsourcing. In *ICML*, pages 603–611, 2016.

[Han *et al.*, 2016] Tao Han, Hailong Sun, Yangqiu Song, Yili Fang, and Xudong Liu. Incorporating external knowledge into crowd intelligence for more specific knowledge acquisition. In *IJCAI*, pages 1541–1547, 2016.

[Ho *et al.*, 2013] Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive task assignment for crowdsourced classification. In *ICML*, pages 534–542, 2013.

[Ipeirotis and Gabrilovich, 2014] Panagiotis G. Ipeirotis and Evgeniy Gabrilovich. Quizz: targeted crowdsourcing with a billion (potential) users. In *WWW*, pages 143–154, 2014.

[Jung, 2014] Hyun Joon Jung. Quality assurance in crowdsourcing via matrix factorization based task routing. In *WWW*, pages 3–8, 2014.

[Kulkarni, 2011] Anand Kulkarni. The complexity of crowdsourcing: Theoretical problems in human computation. In *CHI Workshop on Crowdsourcing and Human Computation*, 2011.

[Li and Liu, 2015] Hongwei Li and Qiang Liu. Cheaper and better: Selecting good workers for crowdsourcing. *Eprint Arxiv*, 2015.

[Liu *et al.*, 2012] Qiang Liu, Jian Peng, and Alexander Ihler. Variational inference for crowdsourcing. In *NIPS*, pages 692–700, 2012.

[Marcus *et al.*, 2015] Adam Marcus, Adam Marcus, Adam Marcus, and Adam Marcus. Argonaut: macrotask crowdsourcing for complex data processing. *Proceedings of the VLDB Endowment*, 8(12):1642–1653, 2015.

[McAllester, 1998] David A. McAllester. Some pac-bayesian theorems. In *COLT*, pages 230–234, 1998.

[Nushi *et al.*, 2015] Besmira Nushi, Adish Singla, Anja Gruenheid, Erfan Zamanian, Andreas Krause, and Donald Kossmann. Crowd access path optimization: Diversity matters. In *HCOMP*, pages 130–139, 2015.

[Raykar *et al.*, 2010] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Machine Learning*, 11(Apr):1297–1322, 2010.

[Roy *et al.*, 2015] Senjuti Basu Roy, Ioanna Lykourentzou, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal*, 24(4):467–491, 2015.

[Salek *et al.*, 2013] M Salek, Y Bachrach, and P Key. Hotspotting - a probabilistic graphical model for image object localization through crowdsourcing. In *AAAI*, pages 1156–1162, 2013.

[Shahaf and Amir, 2007] Dafna Shahaf and Eyal Amir. Towards a theory of ai completeness. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, pages 150–155, 2007.

[Snow *et al.*, 2008] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *EMNLP*, pages 254–263, 2008.

[Tran-Thanh *et al.*, 2013] Long Tran-Thanh, Matteo Venanzi, Alex Rogers, and Nicholas R Jennings. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *AAMAS*, pages 901–908, 2013.

[Venanzi *et al.*, 2014] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *WWW*, pages 155–164, 2014.

[Wang and Zhou, 2016] Lu Wang and Zhihua Zhou. Cost-saving effect of crowdsourcing learning. In *IJCAI*, pages 2111–2117, 2016.

[Wing, 2008] Jeannette M Wing. Five deep questions in computing. *CACM*, 51(1):58–60, 2008.

[Yu *et al.*, 2017] Han Yu, Chunyan Miao, Yiqiang Chen, Simon Fauvel, Xiaoming Li, and Victor R Lesser. Algorithmic management for improving collective productivity in crowdsourcing. *Scientific Reports*, 7(1):12541, 2017.

[Zhou *et al.*, 2012] Dengyong Zhou, John C. Platt, Sumit Basu, and Yi Mao. Learning from the wisdom of crowds by minimax entropy. *NIPS*, 3:2195–2203, 2012.

[Zhou *et al.*, 2015] Dengyong Zhou, Qiang Liu, John C. Platt, Christopher Meek, and Nihar B. Shah. Regularized minimax conditional entropy for crowdsourcing. *Eprint Arxiv*, 2015.