Efficient Online Crowdsourcing with Complex Annotations

Reshef Meir[1,*], **Viet-An Nguyen**[2], **Xu Chen**[2], **Jagdish Ramakrishnan**[2], **Udi Weinsberg**[2]

[1]Technion—Israel Institute of Technology
[2]Central Applied Science, Meta
reshefm@dds.technion.ac.il, {vietan, xuchen2, jagram, udi}@meta.com

Abstract

Crowdsourcing platforms use various truth discovery algorithms to aggregate annotations from multiple labelers. In an online setting, however, the main challenge is to decide whether to ask for more annotations for each item to efficiently trade off cost (i.e., the number of annotations) for quality of the aggregated annotations. In this paper, we propose a novel approach for general complex annotation (such as bounding boxes and taxonomy paths), that works in an online crowdsourcing setting. We prove that the expected average similarity of a labeler is linear in their accuracy *conditional on the reported label*. This enables us to infer reported label accuracy in a broad range of scenarios. We conduct extensive evaluations on real-world crowdsourcing data from Meta and show the effectiveness of our proposed online algorithms in improving the cost-quality trade-off.

Introduction

Crowdsourcing refers to a broad collection of cost-efficient methods to acquire information from a large population of non-experts (Doan, Ramakrishnan, and Halevy 2011). Within crowdsourcing, a common task is to ask *workers* (aka *reviewers* or *labelers*) to provide a correct *annotation* (aka *label*) to a piece of information.

The annotations can be simple such as a yes/no answer to whether there is a car in a given photo, or a real number representing the future price of a commodity. However, in many crowdsourcing tasks, the responses from labelers may comprise of more *complex annotations* such as textual spans, bounding boxes, taxonomy paths, or translations.

These annotations are then aggregated to obtain the best possible estimation of some underlying correct answer. One typical approach to aggregate multiple annotations is to identify good labelers based on custom, domain-specific statistical models (see Related Work). While these specialized models were shown to be effective for their respective tasks, one key limitation for such approach is that a custom algorithm is needed for each type of complex annotation. Recent work has explored pairwise similarities between annotations as a general approach for identifying good labelers across different types of complex annotations (Braylan et al. 2023; Meir et al. 2023). The assumption that good workers are similar to one another in terms of their reported annotations (whereas poor workers do not) is referred to as the *Anna Karenina principle* in Meir et al. (2023).

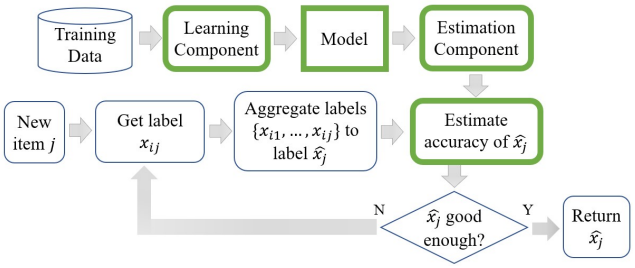

Figure 1: General online crowdsourcing process, in which the components with green solid frame are this work's focus.

Challenges and Contributions

The focus of this paper is on a prevalent and practical crowdsourcing scenario where on the one hand, annotations can be *complex*, and on the other hand we must decide *on-the-fly* whether we should get an additional annotation for each item. In this setting, unfortunately, most of the truth discovery algorithms (mentioned above and surveyed in later sections) are inapplicable. We next briefly describe the online crowdsourcing process of interest to articulate the challenge.

From offline to online crowdsourcing The stylized truth discovery model often used in crowdsourcing literature assumes the existence of an *a priori* given set of questions/items to annotate and a fixed group of workers, each assigned to annotate all or some of the items. The designed algorithm takes these observed annotations as offline input and outputs the best aggregated annotation for each item.

However, practical real-time crowdsourcing processes exhibit a distinct setup. As illustrated in Fig. 1, items arrive sequentially and the system determines when to stop collecting labels for each item. This decision can be informed by the annotations collected so far for this item, and by the knowledge learned from previous annotations for other items used as "training data". The CLARA system developed at Meta

[*]Work done while visiting Meta's Central Applied Science

tackles the same online crowdsourcing problem but only supports categorical labels (Nguyen et al. 2020). *Task assignment*, which selects labelers to annotate items, is an active area of research (Li et al. 2016; Hettiachchi, Kostakos, and Goncalves 2022), but is out of the scope of this work, as it is addressed by a different system.

Main contributions Based on the Anna Karenina (AK) principle as explained above, we suggest an *Online AK algorithm* (OAK) that estimates the accuracy of each labeler by measuring the average similarity to all others on training data. OAK is an adaptation of the Proximity-based Truth Discovery (PTD) algorithm proposed by Meir et al. (2023) for the online crowdsourcing setting with general complex annotations. The main difference is the way average similarity is *used*, which is to decide on when to stop acquiring labels, rather than how to weigh collected labels.

Our main contribution is a *Partition-based extension* of OAK (POAK). Although POAK can be viewed as employing multiple instances of OAK for each reported label type, it offers a more effective means of handling dependencies within the data. From a theoretical standpoint, POAK deviates from the independence assumptions which underpin the AK principle theory presented in Meir et al. (2023). Therefore to provide theoretical foundations for the POAK algorithm, we establish a stronger version of the AK principle which encompasses per-reported-type estimations. From an empirical perspective, we show that the proposed algorithm substantially improves the cost-accuracy trade-offs compared with the baselines on several real-world datasets from various domains collected at Meta. We also propose a third variant, POAKI, which reduces the number of latent variables by incorporating item response theory (IRT) (Baker and Kim 2004).

A full version of the paper with additional results can be found online (Meir et al. 2024).

Related Work

Competence estimation Given that crowdsourcing workers may possess vastly different capabilities due to differences in their inherent competence, training, or effort, it is crucial for crowdsourcing models to learn and account for worker accuracy in order to enhance ground truth estimation (Ipeirotis et al. 2014; Zheng et al. 2017).

In its simplest form, worker competence is captured by a single real number which represents the ability that each worker correctly answers a task (Whitehill et al. 2009; Karger, Oh, and Shah 2011). For categorical labels, one common way to characterize workers' performance is using a confusion matrix to model their ability to correctly label items of different categories (Dawid and Skene 1979; Raykar et al. 2010; Kim and Ghahramani 2012). Another line of work uses a multidimensional vector to model the diverse skill sets of each worker (Welinder et al. 2010; Zhou et al. 2012; Ma et al. 2015).

Naturally, estimation in the above approaches employs statistical analysis that assumes specific label structure—typically multiple-choice questions or a real-valued number—and usually also a particular noise model.

Average similarity The idea of using average similarity of workers as a rough proxy for their competence had also been analyzed in specific domains and applied in practice in *Games with a Purpose* (Von Ahn and Dabbish 2008). Benefits of the average similarity approach were demonstrated theoretically and empirically in specific offline domains including abstractive-summarization models (Kobayashi 2018) and binary labels (Kurvers et al. 2019).

Complex annotations As discussed above, the literature on aggregating complex annotations consists of many task-specific specialized models. Nguyen et al. (2017) propose a HMM-based model for aggregating sequential crowd labels, which was applied to named-entity recognition and information extraction applications. Lin, Mausam, and Weld (2012) introduce LAZYSUSAN to infer the correct answer of crowdsourcing tasks that can have a countably infinite number of possible answers. Branson, Van Horn, and Perona (2017) propose a model for several complex domains. Various other custom models were also proposed for different crowdsourcing applications such as co-reference (Paun et al. 2018; Li, Takamura, and Ananiadou 2020), sequence (Rodrigues, Pereira, and Ribeiro 2014), and translation (Zaidan and Callison-Burch 2011).

Recent work leverages pairwise similarity as a general abstraction for complex annotations. The multidimensional annotation scaling (MAS) algorithm (Braylan and Lease 2020) embeds the pairwise distances in a different space and estimates the competences that maximize the likelihood of observed distances. In a followup paper, the authors suggest a general way to aggregate complex labels (Braylan and Lease 2021). Kawase, Kuroki, and Miyauchi (2019) apply graph algorithms to find the 'core' of the similarity graph, which is assumed to contain the most competent workers. The simplest approach proposed by Meir et al. (2023) was shown to perform consistently well compared to the other approaches in an offline setting.

Online crowdsourcing The online crowdsourcing process that we focus on has been studied under different names in the literature including repeated labeling (Dai et al. 2013; Ipeirotis et al. 2014; Lin et al. 2014), adaptive stopping (Abraham et al. 2016), and incremental relabeling (Drutsa et al. 2020a,b), in which various sophisticated algorithms were developed to guide the relabeling process. However, all of these methods, including the recent CLARA system developed at Meta (Nguyen et al. 2020, 2022), only support simple categorical labels. To the best of our knowledge, our paper is the first to study a general approach for online crowdsourcing with general annotations.

Preliminaries

Labels A label is an element of some predefined set $\mathcal{X}$, which can be either finite or infinite in nature. The most common problems are either categorical, where $\mathcal{X}$ is some finite set of exclusive alternatives (e.g. male/female, types of fruits, names of authors, etc.); or real-valued, where $\mathcal{X}$ are real numbers (representing temperature, price, etc). In this paper, 'annotation' and 'label' are used interchangeably.

Similarity The relation among different possible labels is captured by a *similarity function* $s : \mathcal{X} \times \mathcal{X} \to [0, 1]$. For example, a commonly used similarity for categorical labels is the Hamming similarity, that is, $s(x, x') = 1$ if $x = x'$ and otherwise 0.

Classic Truth Discovery

In a classic truth discovery problem, input is given by a (possibly partial) $n \times m$ table X, where $x_{ij} \in \mathcal{X}$ is the label reported for item $j \leq m$ by worker $i \leq n$. In addition, each item j has a true answer $z_j^* \in \mathcal{X}$. A *classic truth discovery algorithm* is essentially a function mapping input tables to a vector of predicted answers $\hat{\boldsymbol{z}} = (\hat{z}_j)_{j \leq m}$.

Evaluation We evaluate the accuracy of an answer z_j by its similarity to z_j^*. The accuracy of $\hat{\boldsymbol{z}}$ is simply the average over all items, i.e.: $ACC(\hat{\boldsymbol{z}}, \boldsymbol{z}^*) := \frac{1}{m} \sum_{j=1}^{m} s(\hat{z}_j, z_j^*)$.

Average similarity and the AK principle Given full input X,[1] the average similarity of worker i is

$$\pi_i := \frac{1}{n-1} \sum_{i' \neq i} \frac{1}{m} \sum_{j \leq m} s(x_{ij}, x_{i'j}). \tag{1}$$

The average similarity can be computed directly from the input data. In contrast, we often assume that each worker has some intrinsic, unobserved competence that determines her accuracy. We define the competence as $c_i := E[s(x_i, z^*)]$, but note that for this to be well-defined, we need a specific noise model—a distribution connecting the ground truth z^* with the worker's label x_i.

For example, a common noise model for binary labels is the *one-coin* model (also known as the binary Dawid-Skene (1979) model), where $x_{ij} = z_j^*$ with some fixed probability p_i, independently for every item. Note that under the one-coin model and Hamming similarity, the competence of each worker i is exactly $c_i = p_i$. For the one-coin model, a linear connection between c_i and $E[\pi_i]$ was shown by Kurvers et al. (2019). In Meir et al. (2023), it is shown that this linearity holds for any label type, exactly or approximately, under a wide range of noise models as per the Anna Karenina principle. Their PTD algorithm is essentially:

1. Calculate π_i for each worker;
2. Apply a linear transformation to get $\hat{c}_i$ from π_i;
3. Get each $\hat{z}_j$ by a weighted aggregation of the labels $(x_{ij})_{i \leq n}$, where weights depend on $(\hat{c}_i)_{i \leq n}$.

Partial data In general, every item may be labelled by a subset of workers. We denote by $N_j := \{i \in N : x_{ij} \text{ exists}\}$ the set of workers labelling item j. Similarly, $M_i := \{j \in M : x_{ij} \text{ exists}\}$ is the set of items labelled by worker i. We also denote $M_{ii'} := M_i \cap M_{i'}$. Note that Eq. (1) only applies for full data. For a general partial matrix, we compute π_i by taking the average similarity over every label in every $M_{ii'}$.

[1] We explain below the extension to partial data.

Notation	Description
N, M	Set of unique workers / items
x_{ij}	Observed label by worker i to item j
$z_j^*, \hat{z}_j$	True / estimated annotation for item j
$c_i, \hat{c}_i$	True / estimated competence of worker i
$\hat{c}_i^0$	Competence estimated from true labels
M_i, M_i^*	Items [with ground truth] labelled by i
$M_{ii'}$	Items labeled by both i and i'
$\overline{m}_i$	Number of pairwise comparisons involving i
$m_i, m_i^*, m_{ii'}$	Size of $M_i, M_i^*, M_{ii'}$
π_i	Average similarity of i to other workers

Table 1: Common notations used in the paper. Lower-case letters n, m represent set sizes, e.g. $m_i^* = |M_i^*|$.

Online Crowdsourcing

In an online setting, there is no predefined set of items and workers. Instead, there is a dynamic pool of workers, and items arrive sequentially. Upon arrival of an item j, we can ask for a label x_{ij} from worker i. In this paper, we assume the decision on which worker to ask is made by a separate system that we have no control over. Therefore, an online truth-discovery system makes the following decisions for each item: (1) aggregate collected labels, (2) estimate the accuracy of individual/aggregated labels, and (3) decide whether we should stop labeling. See Fig. 1 for a diagram of the labeling process.

Due to constraints on labeling capacity, in most practical tasks the number of collected labels per item is at most 3, and thus aggregation is rather straightforward. Therefore, our goal in this paper is to obtain a good estimation of the label accuracy. Moreover, we will focus on the first decision point as it is most crucial in our crowdsourcing tasks.

Cost-quality trade-off The main implication is that the task reduces to estimating label quality given available information. The performance is then measured by the system cost-quality trade-off: how many labels per item are needed to reach a certain level of quality. An example trade-off curve for the POAK algorithm is presented in Fig. 2. The selection of threshold is a case-by-case decision that hinges on labeling costs and the acceptable margin of error (Nguyen et al. 2020).

Auditor labels Some items may arrive with an 'auditor label', that we can think of as the ground truth. The decision on whether to ask for an auditor label on a particular item is done independently by a different system, and we assume these labeled items are a random sample from all items. We denote by M the set of all items, and by $M^* \subseteq M$ the items for which an auditor label is available. The auditor label is denoted by z_j^* (i.e. it is considered to be the ground truth).

Online algorithms An online accuracy estimation algorithm is composed of two components (see Fig. 1):

A learning component that gets as its input a partial matrix X, possibly with some auditor labels, and outputs a model Θ.

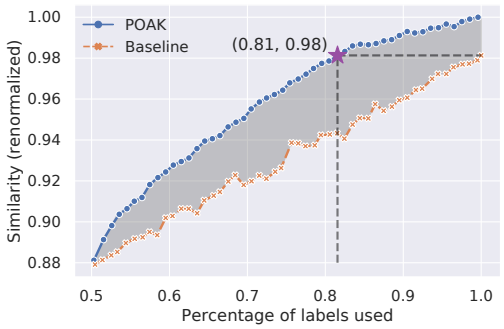

Figure 2: Performance of the POAK algorithm on the Keypoints dataset, compared to a baseline that decides randomly on how many labels to use. Each point on the curve corresponds to the percentage of labels used and associated similarity at a given accuracy threshold. The star marker indicates that POAK achieves on-par accuracy with the baseline only using 81% of the labels. The shaded area (relative AUC) measures the improvement over the baseline.

An estimation component that gets as input a set of reported labels for a particular item $(x_{ij})_{i \in N_j}$ and has access to the model Θ. It outputs an aggregated answer $\hat{z}_j$ with its estimated accuracy $\hat{C}_j$.

Online Anna Karenina Algorithm

We start with a simple model that only includes the estimated competence of each worker. Note that a straightforward way to estimate the competence, is to consider the average accuracy over items with auditor label: $\hat{c}_i^0 := \frac{1}{|M_i^*|} \sum_{j \in M_i^*} s(x_{ij}, z_j^*)$. Clearly, if M_i^* is sampled randomly from M, then $\hat{c}_i^0$ is an unbiased estimator of c_i. However, typically auditor labels are expensive and hence M_i^* is small or empty.

The Learning Component

The first step is to calculate π_i for each worker. Note that under a full input matrix, all workers are treated the same as in Eq. (1). However in practice a worker may have co-labelled many items with some workers, and just a few with others, resulting in noisy pairwise similarity with the latter group. We therefore compute

$$\pi_i := \frac{1}{\overline{m}_i} \sum_{i' \neq i} \sum_{j \in M_i} s(x_{ij}, x_{i'j})$$

where $\overline{m}_i := \sum_{i' \neq i} m_{ii'}$, that is, we average over all pairs of i's label and another label of the same item.

Calibration and semi-supervised learning While π and c are positively correlated, there may be a better linear transformation between them than simply using identity. Indeed, if we have some additional statistical assumptions on the data we can analytically derive such a transformation (Kurvers et al. 2019; Meir et al. 2023), but having access to a small amount of audited labels allows us to take an easier and more general approach.

Algorithm 1: OAK (Learning Component)

Input: dataset X
Output: model Θ
for $i \in N$ **do**
  Set $\overline{m}_i := \sum_{i' \neq i} |M_{ii'}|$
  Set $\pi_i := \frac{1}{\overline{m}_i} \sum_{i' \neq i} \sum_{j \in M_i} s(x_{ij}, x_{i'j})$
  Set $\hat{c}_i^0 := \frac{1}{|M_i^*|} \sum_{j \in M_i^*} s(x_{ij}, z_j^*)$
end
Find the best linear transformation L from π to c^0 using weighted linear regression
For each $i \in N$, set $\hat{c}_i := L(\pi_i)$
return $\Theta := (\hat{c}_i, \overline{m}_i)_{i \in N}$

Recall that we have our preliminary competence estimation $\hat{c}^0$ that is based on the supervised data. While on their own they may be too noisy, we can use them to calibrate the competence estimation, by computing the best linear transformation L from $\pi = (\pi_i)_{i \in N}$ to $(\hat{c}_i^0)_{i \in N}$. This transformation has only two latent variables (slope and intercept) so even a small amount of supervised data is sufficient. If auditor labels are available, we can also combine $\hat{c}_i^0$ and π_i to get a better estimate of c_i. While this aspect is crucial in practice, it is also relatively straightforward, so we defer the details to the Appendix of the full version. We summarize the steps of the learning component of our OAK algorithm in Alg. 1.

The Estimation Component

Estimation in OAK is rather straightforward. Suppose that we have a new item j with labels $(x_{ij})_{i \in N_j}$. First, the algorithm calculates the current estimated label by aggregating all current labels $\hat{z}_j := agg((x_{ij})_{i \in N_j})$. Recall that the aggregation function is decided up front. Then the algorithm predicts the accuracy of $\hat{z}_j$ (denoted $\hat{C}_j$) using the model Θ. This estimation is then compared to a pre-defined threshold; see Fig. 1.

Estimating accuracy of a single label The most important part is to estimate the accuracy of the first label, since asking for another label will at least double the cost. Here aggregation is trivial, as $\hat{z}_j = x_{ij}$ for the first worker $i \in N_j$.

The easiest estimation is just by setting $\hat{C}_j := \hat{c}_i$, i.e. relying completely on the estimated competence of the reporting worker. To deal with small samples we apply additive smoothing, which shrinks the estimation towards the mean accuracy of the entire population, defined as $\bar{c} := \frac{1}{m} \sum_{i \in N_j} m_i \hat{c}_i$, see Alg. 2. The γ is a meta-parameter, set to 10 by default.

Aggregated labels Estimating the accuracy of an aggregated label is tricky. A naive approach that returns the estimated accuracy of one worker only is missing important information: if the other reported labels are identical or similar, this is a strong positive signal, whereas if we know that other workers reported different labels this may suggest lower accuracy. The solution is detailed in the full version.

Algorithm 2: OAK (ESTIMATION COMPONENT)

Input: Labels $(x_{ij})_{i \in N_j}$, model Θ

Output: Estimated label $\hat{z}_j$, estimated accuracy $\hat{C}_j$

Aggregate labels $\hat{z}_j := agg((x_{ij})_{i \in N_j})$

Find closest worker $i^* := \arg\max_{i \in N_j} s(x_{ij}, \hat{z}_j)$

Calculate $\hat{C}_j := \frac{\overline{m}_{i^*}}{\overline{m}_{i^*}+\gamma}\hat{c}_{i^*} + \frac{\gamma}{\overline{m}_{i^*}+\gamma}\bar{c}$

return $(\hat{z}_j, \hat{C}_j)$

Algorithm 3: POAK (ESTIMATION COMPONENT)

Input: Labels $(x_{ij})_{i \in I_j}$, model $\Theta = (\Theta^{(\ell)})_{\ell \le k}$

Output: Estimated label $\hat{z}_j$, estimated accuracy $\hat{C}_j$

Aggregate labels $\hat{z}_j := agg((x_{ij})_{i \in I_j})$

Set ℓ such that $\hat{z}_j \in \mathcal{X}^{(\ell)}$

Find closest worker $i^* := \text{argmax}_{i \in N_j} s(x_{ij}, \hat{z}_j)$

Calculate $\hat{C}_j := \frac{\overline{m}^{(\ell)}_{i^*}}{\overline{m}^{(\ell)}_{i^*}+\gamma}\hat{c}^{(\ell)}_{i^*} + \frac{\gamma}{\overline{m}^{(\ell)}_{i^*}+\gamma}\hat{c}_i$

return $(\hat{z}_j, \hat{C}_j)$

Algorithm Complexity

On time complexity, we need to calculate pairwise similarity between labels within an item for all items. Given that each item contains only a few labels, the time complexity is linear in number of items $\mathcal{O}(M)$, or total number of labels $\mathcal{O}(\sum_i \sum_{i'} |M_{ii'}|)$. Regarding memory complexity, we need to store the estimated confidence of each labeler per annotation type which is $\mathcal{O}(kN)$. As both the time and memory complexities are linear, we believe that our algorithm can be implemented efficiently with little resource concern.

Annotation Types

To demonstrate the main issue we tackle in this paper, consider a population with two types of workers and two equally-frequent labels: type a always identify label A correctly, and type b always identify label B correctly. Each worker makes mistakes on the other label with probability 0.5. Assuming equal priors, each worker has an overall accuracy of $c_i = 0.75$, and every worker has an expected average similarity of $E[\pi_i] = (0.75 + 0.5)/2 = 0.625$.[2] In particular OAK is not able to distinguish between workers since both are equally competent, and can only assess the accuracy of the first label as 0.625, regardless of the label or the identity of the worker. However if the reported label is B and we know that the worker is of type a, then we could tell for sure that the label is correct. In contrast, if the worker is of type b then we know the expected accuracy is only $2/3$.

A different reason that can cause a similar problem is label frequency. Suppose all workers are correct with probability $c_i = 2/3$ regardless of the label, and label A is five times more frequent than B. Clearly, a reported label B is much less likely to be correct than a reported label A (the posteriors are $2/7$ vs. $10/11$ respectively). Yet the simple OAK algorithm will predict the same accuracy in both cases.

[2]Since in case the label matches the type agreement is 1 with her own type and 0.5 with the other type (overall 0.75) and if label mismatches type then agreement is 0.5 with any other worker.

Partition-based OAK Algorithm

We propose a general approach to deal with the above issues without explicitly assuming the underlying model or priors. Our approach is based on a conditional application of the AK principle. We will first describe the modified algorithm, and then explain the theoretical justification. In short, we first partition the space of labels into k types $\mathcal{X} = \uplus_{\ell=1}^{k} \mathcal{X}^{(\ell)}$.

The partition itself is decided externally using domain specific knowledge, where the guiding principle is that similar labels should be grouped together. For example, if there are few categorical labels then each category is a separate type; and if annotations are free-form sentences, the type can be determined by some syntactic feature of the label or some classification of the words within. Alternatively, it is feasible to employ clustering algorithms to automatically determine label types.

Building upon this partitioning idea, instead of assigning a single latent variable per worker to represent their accuracy, we assign k latent variables for each worker, where $c_i^{(\ell)} := E[s(x_i, z^*)|x_i \in \mathcal{X}^{(\ell)}]$ is i's conditional accuracy for label ℓ. Note that we condition on the reported label rather than the true label.

The learning component of the POAK algorithm applies Alg. 1 on each $\mathcal{X}^{(\ell)}$ separately, in order to get a conditional average similarity $\pi_i^{(\ell)}$ and conditional accuracy estimate $\hat{c}_i^{(\ell)} = L^{(\ell)}(\pi_i^{(\ell)})$ for every worker $i \in N$ and label type $\ell \le k$. The new, larger model will then contain $(\hat{c}_i^{(\ell)}, \overline{m}_i^{(\ell)})_{i \in N, \ell \le k}$. Note that there is no explicit estimation of labels' priors or noise model parameters. In its estimation component, the POAK algorithm picks the type ℓ of the aggregated label, and sets the estimated accuracy $\hat{C}_j$ according to $\hat{c}_i^{(\ell)}$, with additive smoothing towards $\hat{c}_i$. See Alg. 3.

Theoretical Justification

To justify the POAK algorithm, we need to show that $\pi_i^{(\ell)}$ is linear in $c_i^{(\ell)}$ in expectation. This is not *a priori* clear, as one of the assumptions in Meir et al. (2023) is that labels from workers are independent conditional on their accuracy—an assumption that is violated once conditioning on the reported label.

We therefore establish an alternative version of the Anna Karenina principle for conditional categorical labels. In particular we provide sufficient conditions to an exact linear relation between $c_i^{(\ell)}$ and $E[\pi_i^{(\ell)}]$, which is positive if (but not only if!) the overall population accuracy is better than random guess.

Statistical model We adopt the general Dawid-Skene model for categorical labels (Dawid and Skene 1979):

- Prior probabilities over labels, denoted by $q^{(\ell)}$ with $\sum_{\ell \le k} q^{(\ell)} = 1$;

- Worker types. A type is specified by a $k \times k$ confusion matrix $\mathcal{M}_i$, where $\mathcal{M}_i^{\tau\to\ell} := Pr[x_i = \ell | z^* = \tau]$. We require $\sum_{\ell \le k} \mathcal{M}_i^{\tau\to\ell} = 1$ for all i, τ;
- Conditional independence among workers: $Pr[x_i = \ell | z^* = \tau, x_{i'}] = \mathcal{M}_i^{\tau\to\ell}$.

Meir et al. (2023) imposes a strong restriction on the model where each $\mathcal{M}_i$ depends only on the accuracy of i, which is a scalar parameter, in order to derive the corresponding linear relation. We make no assumption on the confusion matrices, and ask how expected conditional similarity behaves as function of the conditional accuracy $c_i^{(\ell)}$.

Denote by $p_i^{(\ell)} := \mathcal{M}_i^{\ell\to\ell}$ the probability of i correctly identifying label ℓ. We define a *partial type* $\mathcal{M}_i^{(-\ell)}$ which is $\mathcal{M}_i$ without the column $\mathcal{M}_i^{\ell\to\cdot}$. Then we can fix the partial type, and ask how both $\pi_i^{(\ell)}$ and $c_i^{(\ell)}$ change as a function of the remaining latent variables, and in particular $p_i^{(\ell)}$. Our main result is the following.

Theorem 1 (Conditional Anna Karenina theorem for categorical data)**.** *Fix prior probabilities q, a category ℓ, and a worker i with partial type $\mathcal{M}_i^{(-\ell)}$. Then there are constants $\alpha^{(\ell)}, \beta^{(\ell)}$ such that $E[\pi_i^{(\ell)}] = \alpha^{(\ell)} c_i^{(\ell)} + \beta^{(\ell)}$.*

We present the key components of the proof here and defer the complete proof to the full version.First, we note that since $\pi_i^{(\ell)}$ is an average over comparisons to random labels reported by a random worker i', we have (omitting the item subscript j):

$$E[\pi_i^{(\ell)}] = E_{i'}[E[s(x_{i'}, x_i) | x_i = \ell]] = E_{i'}[Pr[x_{i'} = \ell | x_i = \ell]].$$

Hence by linearity of expectation, it is sufficient to show the following proposition.

Proposition 2. *$Pr[x_{i'} = \ell | x_i = \ell]$ is linear in $c_i^{(\ell)}$, for any worker type $\mathcal{M}_{i'}$.*

Proof. We split to cases when x_i agrees or disagrees with the truth z^*, and show that both terms are linear in $c_i^{(\ell)}$:

$$\begin{aligned} Pr[x_{i'} = \ell | x_i = \ell] &= Pr[x_{i'} = \ell | z^* = \ell] c_i^{(\ell)} \qquad (2)\\ &\quad + \sum_{\tau \neq \ell} Pr[x_{i'} = \ell | z^* = \tau] Pr[z^* = \tau | x_i = \ell] \\ &= p_{i'}^{(\ell)} c_i^{(\ell)} + \sum_{\tau \neq \ell} M_{i'}^{\tau\to\ell} \frac{q^{(\tau)} M_i^{\tau\to\ell}}{Pr[x_i = \ell]} \qquad (3)\\ &= p_{i'}^{(\ell)} c_i^{(\ell)} + \frac{1}{Pr[x_i = \ell]} \sum_{\tau \neq \ell} q^{(\tau)} M_i^{\tau\to\ell} M_{i'}^{\tau\to\ell} \qquad (4) \end{aligned}$$

The first term in Eq. (4) is obviously linear in $c_i^{(\ell)}$ since $\mathcal{M}_{i'}$ is fixed. For the second term, we first observe that each $q^{(\tau)} M_i^{\tau\to\ell} M_{i'}^{\tau\to\ell}$ for $\tau \neq \ell$ is completely determined by the fixed terms $q, \mathcal{M}_{i'}$ and $\mathcal{M}_i^{(-\ell)}$. It remains to show that $\frac{1}{Pr[x_i = \ell]}$ is linear in $c_i^{(\ell)}$. Note that

$$c_i^{(\ell)} = \frac{q^{(\ell)} p_i^{(\ell)}}{Pr[x_i = \ell]} \qquad (5)$$

$$Pr[x_i = \ell] = q^{(\ell)} p_i^{(\ell)} + \sum_{\tau \neq \ell} q^{(\tau)} \mathcal{M}_i^{\tau\to\ell} \qquad (6)$$

$$\forall Y, Z > 0,\ \frac{1}{Z+Y} = \frac{-1}{Y} \cdot \frac{Z}{Z+Y} + \frac{1}{Y} \qquad (7)$$

Finally, denote $Z := q^{(\ell)} p_i^{(\ell)}$; $Y = \sum_{\tau \neq \ell} q^{(\tau)} \mathcal{M}_i^{\tau\to\ell}$, and note that Y is a constant. We have

$$\frac{1}{Pr[x_i = \ell]} = \frac{1}{Z+Y} = \frac{-1}{Y} \cdot \frac{Z}{Z+Y} + \frac{1}{Y} = \frac{-1}{Y} \cdot c_i^{(\ell)} + \frac{1}{Y},$$

as required. □

Remark 3. The higher $\alpha^{(\ell)}$ is w.r.t $\beta^{(\ell)}$, the better we can separate good workers from poor ones. Note that a sufficient condition for $\alpha^{(\ell)}$ to be positive, is that $E_{\mathcal{M}_{i'}}[p_{i'}^{(\ell)}]$ is higher than $E_{\mathcal{M}_{i'}}[\mathcal{M}_{i'}^{\tau\to\ell}]$ for all $\tau \neq \ell$. I.e., that the overall competence to identify ℓ is higher than the overall chance to incorrectly report a label as ℓ.

Item Response Theory

Item Response Theory is a statistical model developed for standardized tests, which posits that the probability of answering a question correctly as a function of three factors of the question:

$$\mathrm{P}^{IRT}(x_{ij}) = p_j^0 + \frac{1 - p_j^0}{1 + \exp(b_j(d_j - c_i))},$$

where d_j, b_j, p_j^0 correspond to item j's difficulty, separation, and base rate; and P^{IRT} is the probability that answer x_{ij} is correct (Baker and Kim 2004). We adopt the concept and the formula but use it in two unusual ways:

- We assign the three 'question factors' mentioned above not to each item, but to each *reported label* type ℓ.
- We do not restrict its usage to categorical data.

This means we define a model with $(3k+n)$ latent variables, compared to $(n \times k)$ for POAK, that captures the expected similarity of x_{ij} to the ground truth, conditional on $x_{ij} \in \mathcal{X}^{(\ell)}$.

In the full version, we describe the POAK-IRT (or POAKI) algorithm, which first runs POAK and then reduce the number of parameters by finding the IRT parameters with the best fit.

We also prove that the common case of single-parameter workers with arbitrary priors on true answers is captured by our IRT model without any loss of precision.

Empirical Results

To showcase the effectiveness of our proposed methods, we conduct extensive numerical experiments on four complex annotation datasets and compare against several baselines.

Method	Conf.-based	Aggregation
POAK-Weight	Yes	Averaging/voting
POAK-BAU	Yes	Selection
OAK-Weight	Yes	Averaging/voting
OAK-BAU	Yes	Selection
SAD	No	Selection
UNIFORM	No	Averaging/voting

Table 2: Summary of key characteristics of competing methods in our experiments.

Dataset	Labelers	Similarity measure	Audit labels
KEYPOINT	~ 50	Gaussian similarity	Yes
TAXONOMY	~ 500	Jaccard	Yes
TREEPATH	~ 70	Hierarchical Hamming	No
BOUNDINGBOX	~ 100	Image Jaccard	Yes

Table 3: Basic information of the real crowdsourcing datasets.

Setup of Numerical Experiments

Baselines and datasets To evaluate the three variants of our proposed algorithms (OAK, POAK, and POAKI), we consider several baseline methods and label aggregation methods (summarized in Tab. 2) including :

- SAD: Smallest Average Distance (Braylan and Lease 2020). For a job with multiple labels, select the label which has the smallest average distance to other labels.
- BAU: Best Available User (Braylan and Lease 2020). Given the (estimated) confidences of multiple labels for a job, select the label with the highest confidence. This aggregation method can be combined with any confidence estimation methods like ours.
- UNIFORM: Uniform averaging or majority voting. This is in contrast to the "weight" method where weights are determined by the estimated labeler confidence.

All methods are applied on four real crowdsourcing datasets obtained from Meta, covering a broad range of different labeling tasks. See basic information of the datasets in Tab. 3. For example, in the TAXONOMY dataset, each annotation is a subset of 26 predefined topics. The similarity of two annotations is their Jaccard similarity and aggregation is performed by majority voting on each topic.

Note that to use the POAK algorithm we need to somehow partition annotations into types. In all datasets we used a simple straightforward partition. E.g. in the TAXONOMY dataset we assign each singleton to a type and group all non-singletons into a separate type.

Comprehensive descriptions and partitioning details are included in the full version.

Evaluation We first randomly split each dataset into a training set and a test set. We then run the learning component (Alg. 1) of each variant to estimate labeler's confidence. Given the learned confidence, we implement the estimation component (Alg. 2) to estimate the accuracy of the first label for each job in the test set, only inquiring the next label if the accuracy is below a given threshold. At each threshold, we calculate the cost and accuracy by averaging over all jobs in the test set, which corresponds to a point on the curve in Fig. 2. By varying the threshold, we are able to generate a cost-accuracy curve for each method as in Fig. 2.

This cost-accuracy curve of each algorithm is compared against the UNIFORM baseline which uses a biased coin-flip to decide whether to use one or all annotations. We evaluate the performance of all methods by computing the *Relative Area Under the Curve* (RAUC), represented as the shaded area in the illustrating example in Fig. 2.

Results

Comparison with the baselines Fig. 3 shows the RAUC results of the different methods on the four datasets.

The confidence-based methods POAK-Weight and OAK-Weight consistently outperform non-model based alternatives, SAD and UNIFORM, which justifies the usefulness of confidence estimation in improving the cost-quality trade-off curve. Averaging or majority voting proves to be more effective than selection across most of the settings. This finding is in line with a similar conclusion from Braylan and Lease (2021).

In addition, POAK outperforms OAK with POAK-Weight dominating all other methods. This implies that the competencies of labelers vary across different annotation types. The partition-based method can effectively learn and adjust for this heterogeneity. Finally, as the training sample size increases, the performance of all methods improves as expected as the methods can make more accurate estimations of labeler confidence.

Deep dive on POAK As POAK-Weight dominates all other alternatives, we dive deeper into the different variants of POAK[3], including POAKI and POAK-IRT, to examine the effectiveness of calibration in confidence estimation. It turns out that data is not uniformly distributed across different annotation types. We tackle this by computing the POAKI estimation *in addition*, using it as a baseline for smoothing instead of OAK. As shown in Fig. 4, both POAKI and POAK-IRT outperform POAK, highlighting the value of calibration particularly when training sample is small. In addition, IRT-based regularization (purple curve) further improves over POAKI by allowing for information sharing across partitions to prevent overfitting.

In addition to the performance evaluation, we also examine the predictive accuracy of POAK. Specifically, the predicted accuracy of labels and their true accuracy are highly correlated (Fig. 5). Furthermore, the estimation accuracy exhibits heterogeneity across different annotation types.

Conclusion

In this paper, we develop novel modeling approaches to improve the efficiency of online crowdsourcing processes

[3]For simplicity, the term "Weight" is omitted from names hereafter, given that all POAK variants use this aggregation method.

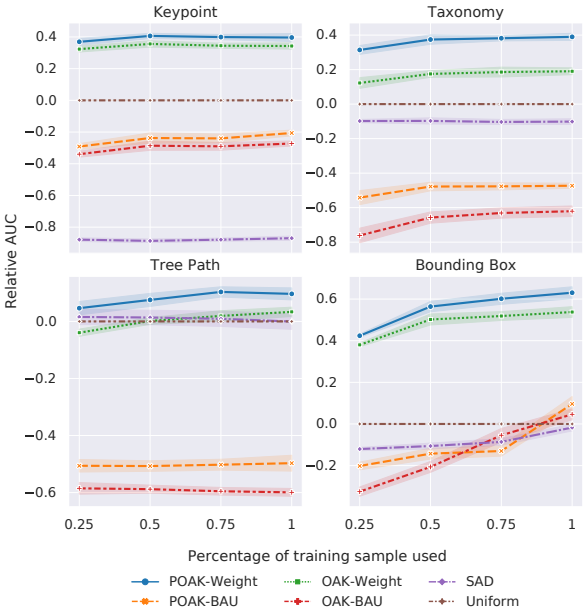

Figure 3: RAUC results (relative to UNIFORM) of all methods on four datasets. Point estimates and 95% confidence intervals are obtained over 10 trails under each setting.

Figure 4: Comparison between different POAK variants.

with complex annotations. The models proposed are task-independent and applicable to any complex annotation tasks in which the pairwise similarity between two arbitrary annotations can be defined. These models are based on the underlying Anna Karenina principle that good workers are similar to one another in their reported annotations.

We first extend previous work on PTD to propose OAK in the online setting and then introduce two extensions: (1) POAK which estimates the accuracy of complex annotations by first partitioning the observed annotations into types and then applying OAK to estimate workers' per-type competence, and (2) POAKI that reduces the number of parameters by using Item Response Theory.

We provide theoretical proofs that the Anna Karenina principle extends to per-reported-type estimations, which generalizes the results of Meir et al. (2023). We also provide extensive empirical results comparing the effectiveness of our methods on four real-world applications.

Figure 5: A plot of estimated accuracy $\hat{c}_i^{(\ell)}$ vs. actual accuracy computed over all items in the test set. Each point represents a pair (i, ℓ) of worker and label type, where larger dots represent pairs with more samples in the data.

Acknowledgements

We appreciate the helpful comments of the anonymous reviewers. Reshef Meir was partly funded by the Israel Science Foundation (ISF grant 2539/20).

References

Abraham, I.; Alonso, O.; Kandylas, V.; Patel, R.; Shelford, S.; and Slivkins, A. 2016. How Many Workers to Ask? Adaptive Exploration for Collecting High Quality Labels. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, 473–482.

Baker, F. B.; and Kim, S.-H. 2004. *Item response theory: Parameter estimation techniques*. CRC press.

Branson, S.; Van Horn, G.; and Perona, P. 2017. Lean crowdsourcing: Combining humans and machines in an online system. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7474–7483.

Braylan, A.; and Lease, M. 2020. Modeling and Aggregation of Complex Annotations via Annotation Distances. In *Proceedings of The Web Conference (WWW)*, 1807–1818.

Braylan, A.; and Lease, M. 2021. Aggregating Complex Annotations via Merging and Matching. In *Proceedings of the ACM Conference on Knowledge Discovery & Data Mining (SIGKDD)*, 86–94.

Braylan, A.; Marabella, M.; Alonso, O.; and Lease, M. 2023. A General Model for Aggregating Annotations Across Simple, Complex, and Multi-Object Annotation Tasks. *Journal of Artificial Intelligence Research (JAIR)*, 78: 901–973.

Dai, P.; Lin, C. H.; Mausam; and Weld, D. S. 2013. POMDP-based control of workflows for crowdsourcing. *Artificial Intelligence*, 202: 52–85.

Dawid, A. P.; and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28: 20–28.

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

Drutsa, A.; Fedorova, V.; Ustalov, D.; Megorskaya, O.; Zerminova, E.; and Baidakova, D. 2020a. Crowdsourcing Practice for Efficient Data Labeling: Aggregation, Incremental

Relabeling, and Pricing. In *Proceedings of the International Conference on Management of Data*, 2623–2627.

Drutsa, A.; Fedorova, V.; Ustalov, D.; Megorskaya, O.; Zerminova, E.; and Baidakova, D. 2020b. Practice of Efficient Data Collection via Crowdsourcing: Aggregation, Incremental Relabelling, and Pricing. In *Proceedings of the International Conference on Web Search and Data Mining (WSDM)*, 873–876.

Hettiachchi, D.; Kostakos, V.; and Goncalves, J. 2022. A survey on task assignment in crowdsourcing. *ACM Computing Surveys (CSUR)*, 55(3): 1–35.

Ipeirotis, P. G.; Provost, F.; Sheng, V. S.; and Wang, J. 2014. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, 28(2): 402–441.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1953–1961.

Kawase, Y.; Kuroki, Y.; and Miyauchi, A. 2019. Graph mining meets crowdsourcing: Extracting experts for answer aggregation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1272–1279.

Kim, H.-C.; and Ghahramani, Z. 2012. Bayesian classifier combination. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 619–627.

Kobayashi, H. 2018. Frustratingly Easy Model Ensemble for Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4165–4176.

Kurvers, R. H.; Herzog, S. M.; Hertwig, R.; Krause, J.; Moussaid, M.; Argenziano, G.; Zalaudek, I.; Carney, P. A.; and Wolf, M. 2019. How to detect high-performing individuals and groups: Decision similarity predicts accuracy. *Science Advances*, 5(11).

Li, G.; Wang, J.; Zheng, Y.; and Franklin, M. J. 2016. Crowdsourced data management: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 28(9): 2296–2319.

Li, M.; Takamura, H.; and Ananiadou, S. 2020. A neural model for aggregating coreference annotation in crowdsourcing. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 5760–5773.

Lin, C. H.; Mausam; Weld, D. S.; et al. 2014. To re (label), or not to re (label). In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 151–158.

Lin, C. H.; Mausam, M.; and Weld, D. S. 2012. Crowdsourcing control: Moving beyond multiple choice. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 491–500.

Ma, F.; Li, Y.; Li, Q.; Qiu, M.; Gao, J.; Zhi, S.; Su, L.; Zhao, B.; Ji, H.; and Han, J. 2015. FaitCrowd: Fine grained truth discovery for crowdsourced data aggregation. In *Proceedings of the Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 745–754.

Meir, R.; Amir, O.; Cohensius, G.; Ben-Porat, O.; Ben-Shabat, T.; and Xia, L. 2023. Frustratingly Easy Truth Discovery. In *Proceedings of the International Conference on Artificial Intelligence (AAAI)*.

Meir, R.; Nguyen, V.-A.; Chen, X.; Ramakrishnan, J.; and Weinsberg, U. 2024. Efficient Online Crowdsourcing with Complex Annotations. arXiv:2401.15116.

Nguyen, A. T.; Wallace, B.; Li, J. J.; Nenkova, A.; and Lease, M. 2017. Aggregating and Predicting Sequence Labels from Crowd Annotations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 299–309.

Nguyen, V.-A.; Shi, P.; Ramakrishnan, J.; Torabi, N.; Arora, N. S.; Weinsberg, U.; and Tingley, M. 2022. Crowdsourcing with Contextual Uncertainty. In *Proceedings of the Conference on Knowledge Discovery & Data Mining*, 3645–3655.

Nguyen, V.-A.; Shi, P.; Ramakrishnan, J.; Weinsberg, U.; Lin, H. C.; Metz, S.; Chandra, N.; Jing, J.; and Kalimeris, D. 2020. CLARA: Confidence of Labels and Raters. In *Proceedings of the Conference on Knowledge Discovery & Data Mining*, 2542–2552.

Paun, S.; Chamberlain, J.; Kruschwitz, U.; Yu, J.; and Poesio, M. 2018. A Probabilistic Annotation Model for Crowdsourcing Coreference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1926–1937.

Raykar, V. C.; Yu, S.; Zhao, L. H.; Valadez, G. H.; Florin, C.; Bogoni, L.; and Moy, L. 2010. Learning From Crowds. *Journal of Machine Learning Research (JMLR)*, 11: 1297–1322.

Rodrigues, F.; Pereira, F.; and Ribeiro, B. 2014. Sequence labeling with multiple annotators. *Machine Learning*, 95(2): 165–181.

Von Ahn, L.; and Dabbish, L. 2008. Designing games with a purpose. *Communications of the ACM*, 51(8): 58–67.

Welinder, P.; Branson, S.; Perona, P.; and Belongie, S. J. 2010. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2424–2432.

Whitehill, J.; Wu, T.-f.; Bergsma, J.; Movellan, J.; and Ruvolo, P. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2035–2043.

Zaidan, O.; and Callison-Burch, C. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, 1220–1229.

Zheng, Y.; Li, G.; Li, Y.; Shan, C.; and Cheng, R. 2017. Truth Inference in Crowdsourcing: Is the Problem Solved? *Proceedings of the VLDB Endowment*, 541–552.

Zhou, D.; Basu, S.; Mao, Y.; and Platt, J. C. 2012. Learning from the Wisdom of Crowds by Minimax Entropy. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2195–2203.