Truth Inference at Scale: A Bayesian Model for Adjudicating Highly Redundant Crowd Annotations

Yuan Li
University of Melbourne
Melbourne, Australia
yuanl4@student.unimelb.edu.au

Benjamin I. P. Rubinstein
University of Melbourne
Melbourne, Australia
brubinstein@unimelb.edu.au

Trevor Cohn
University of Melbourne
Melbourne, Australia
tcohn@unimelb.edu.au

ABSTRACT

Crowd-sourcing is a cheap and popular means of creating training and evaluation datasets for machine learning, however it poses the problem of 'truth inference', as individual workers cannot be wholly trusted to provide reliable annotations. Research into models of annotation aggregation attempts to infer a latent 'true' annotation, which has been shown to improve the utility of crowd-sourced data. However, existing techniques beat simple baselines only in low redundancy settings, where the number of annotations per instance is low (≤ 3), or in situations where workers are unreliable and produce low quality annotations (e.g., through spamming, random, or adversarial behaviours.) As we show, datasets produced by crowd-sourcing are often not of this type: the data is highly redundantly annotated (≥ 5 annotations per instance), and the vast majority of workers produce high quality outputs. In these settings, the majority vote heuristic performs very well, and most truth inference models underperform this simple baseline. We propose a novel technique, based on a Bayesian graphical model with conjugate priors, and simple iterative expectation-maximisation inference. Our technique produces competitive performance to the state-of-the-art benchmark methods, and is the only method that significantly outperforms the majority vote heuristic at one-sided level 0.025, shown by significance tests. Moreover, our technique is simple, is implemented in only 50 lines of code, and trains in seconds. [1]

KEYWORDS

truth inference, crowdsourcing, Bayesian methods, probabilistic models

[1]The code and the datasets are made available at https://github.com/yuan-li/truth-inference-at-scale.

1 INTRODUCTION

Large volumes of labelled data are crucial for machine learning, retrieval and search, both for training supervised methods and for evaluation. Ideally, ground truth labels are collected from experts, but such sources are often too slow or costly, motivating the use of data annotation by crowdsourcing [2, 7]. This has been shown to be highly effective for tasks which are difficult for machines to automate but relatively easy for humans, such as document categorization, image annotation, and sentiment analysis. Using non-expert labels for training machine learning algorithms can be as effective as using gold standard annotations from experts [16, 21], despite being much cheaper and faster to obtain.

Unfortunately, crowdsourced labels are often noisy or biased, which may be due to workers' lack of expertise, workers cheating or otherwise not performing the annotation task correctly [6]. This problem can be alleviated by filtering workers based on their profile and working history. However, this problem is not fully solved by simple quality control mechanisms currently provided by crowdsourcing platforms; consequently, most crowd data studies collect several labels for each item from different workers, which are aggregated post-hoc to infer an accurate consensus annotation.

Both the average number of labels per item (#labels/#items), and the quality of worker labels, are important factors in the performance of aggregation methods. Low-redundancy (e.g. #labels/#items ≤ 3) and/or low-quality scenarios (e.g. there are adversarial workers) are challenging and have attracted much attention. Such scenarios require methods to make full use of a few available labels by being able to interpret labels from different workers. Models which incorporate per-worker confusion matrices [3, 10, 18, 22] are flexible enough to capture different types of worker behaviour, and have shown to be effective in low-redundancy and/or low-quality scenarios.

However, the simplest aggregation technique, majority voting (MV), still predominates in practice. MV is an ensemble where every worker is given equal vote, based on the assumption that with increasing redundancy, the most common label will converge to the true label. Despite its simplicity, on many real-world crowdsourced datasets, MV outperforms most models of truth inference, as we show in our experimental evaluation, and the best method from prior work only outperforms MV by a small margin. To explain this observation, and the continued popularity of MV, we identify several misalignments between the experimental setup in previous papers and real-world scenarios:

Redundancy & worker quality. In practice #labels/#items is usually 5+ to ensure sufficient redundancy, and crowdsourcing platforms provide mechanisms to select high quality workers (e.g. in the collection of question answering dataset

SQuAD [17], workers were filtered based on the amount and acceptance rate of their prior work history). In such settings, MV is very competitive. Worryingly, our results show that confusion-matrix-based methods are inferior to MV in terms of mean accuracy across datasets, despite their utility in low-resource and/or low-quality scenarios.

Scale. Several datasets used in prior work were small with respect to the number of labels collected from crowd (#labels). But on large-scale datasets, like senti used in our experiment with 569k labels, the scalability of inference becomes critical. For example, GLAD [26], achieving the third highest mean accuracy among all existing methods in our experiment, takes more than 3 hours to train, while MV runs in 3.5 seconds.

This work is also motivated by qualitative considerations: ease of understanding, implementation, and hyper-parameter setting, for which MV is compelling, being parameter free. However, empirically the accuracy of MV is far from perfect, even for highly redundantly annotated datasets. We seek accurate, fast solutions to the high-resource high-quality scenario that are easy to implement. Our method is one such solution.

This paper presents a novel method for aggregating crowd-sourced labels for classification. We propose a generative Bayesian model, relaxing a discrete binary problem to a continuous regression task, which we model using a normal likelihood and conjugate inverse-Gamma prior. Our method allows for straightforward inference using expectation-maximisation (EM; [5]), which is implemented in 50 lines of code (see Appendix A). The resulting update rules reveal a novel criterion for learning from crowds—namely the idea of minimizing the likelihood that "every worker is making mistakes".

Our base model has interpretable parameters and intuitive inference rules, which makes it easy to understand and use. We extend our base binary model in three ways: estimating different weights for each label, to capture both the reliability of a worker votes in favour and against an item belonging to a class; a means of correcting for model deficiency; and employing the one-versus-rest strategy to allow the model to operate in a multi-class setting. Our experimental results show our proposed methods achieve the highest mean accuracy over 10 real-world datasets drawn from several different application areas, with different sizes and differing degrees of label redundancy, and a total running time on the same datasets second only to MV.

2 RELATED WORK

Ever since Dawid and Skene [3] proposed a model (DS) to aggregate the clinical diagnoses of doctors, various methods have been proposed to aggregate worker labels and infer true annotations. The DS model uses the confusion matrix to capture the probabilities that a worker label is generated for an item conditioned on the item's true annotation. Extending the DS model by adding Dirichlet priors for the values in the confusion matrix, Kim and Ghahramani [10] then integrated the confusion matrix parameters out and used Gibbs sampling for inference. Simpson et al. [20] developed variational Bayesian inference for the model proposed in [10], which is computationally efficient and outperforms the Gibbs sampling method.

Furthermore, they analysed the clusters naturally formed by the inferred workers' confusion matrices. Venanzi et al. [22] assumed there are a fixed number of clusters of workers, with workers in the same cluster possessing similar confusion matrices instead of modelling individual confusion matrices as done in previous works. In this way worker clusters are formed at inference-time instead of post-inference. Adding a Dirichlet process to the model to generate the Dirichlet priors on worker confusion matrices, Moreno et al. [14] also allowed workers to generate their own confusion matrices from the cluster's confusion matrix in order to increase model flexibility.

Unlike the above works, Whitehill et al. [26] didn't use confusion matrices, but borrowed ideas from Item Response Theory (IRT) and proposed an unsupervised version of previous IRT models. These estimate not only the true annotations and worker ability but also the difficulty of items. Zhou et al. [29] also developed an algorithm estimating the true annotations by using a minimax entropy principle, which assumes there are categorical distributions per worker-item pair and the worker labels are generated from those categorical distributions. The objective there is in finding the true annotations that minimize the maximum entropy of those categorical distributions. Such categorical distributions are more concentrated on worker labels and can be roughly considered as having higher likelihood of generating labels.

The database community has also independently studied the problem of resolving conflicts between information from different sources. Demartini et al. [4] developed a model similar to the DS model that also used EM but just modelled a single accuracy parameter instead of a full confusion matrix. Li et al. [12] adopted the Gaussian to model error between the truth and worker label, which is similar to our model but not a probabilistic model. They developed an iterative approach which alternatively estimated the truth based on variances, and set the variances to be the upper-bounds of their 95% confidence interval based on the estimated truth. Aydin et al. [1] adopted a two-step procedure that iteratively updates estimated truth and worker weights so that the sum of worker weight times the distance between worker label and estimated truth is minimized.

3 PROPOSED MODELS

In this section, we first define the crowdsourced annotation aggregation problem, then describe our proposed model, Bayesian Weighted Average (BWA), for binary classification tasks, followed by an extension to multi-class classification tasks, and strategies to set its hyper-parameters.

Problem statement. We assume that a task contains a number of homogeneous *instances* or *items*, where each item has its own *true label*. The true labels are drawn from a *label set*. All instances are uploaded to a crowdsourcing platform by requesters. Generally the requesters don't know the true labels, so they ask *workers* to label them. Workers need to choose a label from the label set per item. Typically a worker only labels a small fraction of all items in a task. The labels collected from workers are called *worker labels*. Finally, aggregation methods are used to infer the true annotations based on all worker labels.

Table 1: Mathematical notation.

Symbol	Description
N	number of items
i	item index, $i \in \{1, \ldots, N\}$
W	number of workers
j	worker index, $j \in \{1, \ldots, W\}$
K	number of classes
k, l	class/label index, $k, l \in \{0, \ldots, K-1\}$
W_i	set of workers who have labelled item i
N_j	set of items that worker j has labelled
Binary setting	
z_i	item i's latent true label (continuous), $z_i \in \mathbb{R}$
y_{ij}	worker j's observed label to item i (discrete), $y_{ij} \in \{0, 1\}$
Multi-class setting	
z_{ki}	score given to item i by model k which classifies items into class-k or not class-k, $z_{ki} \in \mathbb{R}$
y_{kij}	binary indicator, equalling 1 if worker j labels item i as class k otherwise 0

3.1 The Bayesian Weighted Average Model

We first consider binary classification tasks. Without loss of generality, let $y_{ij} \in \{0, 1\}$ be worker j's label to item i and let $z_i \in \mathbb{R}$ represent item i's true label. We model z_i's as continuous variables to make tractable the following optimization problem for inferring the truth, Z,

$$Z^* = \arg\max_Z P(Z|Y, \alpha), \tag{1}$$

where Z and Y are collections of their lowercase variables, α denotes all hyper-parameters, $P(Z|Y, \alpha)$ is the posterior distribution for Z, and Z^* is the maximum a posteriori (MAP) estimate of Z. Further discussion of our modelling choices is provided in Section 4.

After Z^* is computed, the discrete true label can be determined by rounding, i.e. the true label is 1 if $z_i^* > 0.5$, otherwise 0. Frequently used symbols are defined in Table 1.

3.1.1 The Generative Process and Joint Distribution. Our proposed models are all probabilistic generative models. We assume z_i is drawn from a Gaussian prior with mean μ and inverse variance λ, and y_{ij} is drawn from a Gaussian distribution with mean z_i and inverse variance v_j. Here v_j is a worker specific value for worker j and every v_j is drawn from Gamma($\frac{a_v}{2}, \frac{b_v}{2}$). The plate diagram for this model is shown in Figure 1.

The joint distribution $P(Z, Y, V|\mu, \lambda, a_v, b_v)$ can be factorized as

$$P(Z, Y, V|\mu, \lambda, a_v, b_v) = P(Z|\mu, \lambda) \cdot P(Y|Z, V) \cdot P(V|a_v, b_v)$$

$$= \prod_i \mathcal{N}\left(z_i \middle| \mu, \lambda^{-1}\right)$$

$$\cdot \prod_i \prod_{j \in W_i} \mathcal{N}\left(y_{ij} \middle| z_i, v_j^{-1}\right) \cdot \prod_j \text{Gamma}\left(v_j \middle| \frac{a_v}{2}, \frac{b_v}{2}\right), \tag{2}$$

where $V = \{v_j\}_{j=1}^W$. λ and v_j model the inverse variance of $z_i - \mu$ and $y_{ij} - z_i$. The larger λ and v_j are, the smaller the corresponding errors
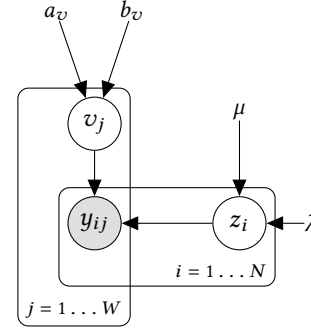
Figure 1: Plate diagram for our proposed BWA model (binary case).

are likely to be. In this way they can be interpreted as encoding worker reliability, a perspective we explore further in Section 3.1.3.

Our proposed BWA model is deficient: data observations y_{ij} can only be binary, while the generative model has continuous support. We could force all y_{ij} to be binary variables ($\{0, 1\}$) then re-normalize the distributions accordingly to address the deficiency issue (Section 4.3). But the linking distribution between y_{ij}, z_i and v_j then becomes a logistic function, which is much more difficult to work with as no conjugate prior for v_j exists.

3.1.2 Inference. In order to find the most likely Z, we first integrate out V in Equation (2) to obtain $P(Z, Y|\alpha)$ where $\alpha = (\mu, \lambda, a_v, b_v)$ are the hyper-parameters.

$$P(Z, Y|\alpha) = \int P(Z, Y, V|\mu, \lambda, a_v, b_v) \, dV$$

$$= \prod_i \mathcal{N}(z_i|\mu, \lambda^{-1})$$

$$\cdot \prod_j \int \prod_{i \in N_j} \mathcal{N}\left(y_{ij}\middle|z_i, v_j^{-1}\right) \cdot \text{Gamma}\left(v_j\middle|\frac{a_v}{2}, \frac{b_v}{2}\right) \, dv_j$$

$$\propto \prod_i e^{-\frac{\lambda}{2}(z_i - \mu)^2} \cdot \prod_j \left(\frac{1}{2}b_v + \frac{1}{2}\sum_{i \in N_j}(z_i - y_{ij})^2\right)^{-\frac{a_v + |N_j|}{2}}.$$

Since $P(Z, Y|\alpha) = P(Z|Y, \alpha)P(Y|\alpha)$ and $P(Y|\alpha)$ is independent from Z, maximising $P(Z, Y|\alpha)$ with respect to Z is equivalent to solving the program (1), i.e.

$$\arg\max_Z P(Z|Y, \alpha) = \arg\max_Z P(Z, Y|\alpha).$$

As we shortly show, λ, a_v, and b_v have intuitive meanings, and accordingly can be set manually, while μ can be optimized easily with Z. Therefore, the goal becomes to maximise $P(Z, Y|\alpha)$ with respect to Z and μ, which is equivalent to minimising the negative log likelihood $-\log P(Z, Y|\alpha)$ with respect to Z and μ:

$$\min_{Z, \mu} \sum_i \frac{\lambda}{2}(z_i - \mu)^2 + \sum_j \frac{a_v + |N_j|}{2} \log\left(b_v + \sum_{i \in N_j}(z_i - y_{ij})^2\right)$$
$$+ const., \tag{3}$$

where *const.* is the sum of values independent from Z and μ, and thus can be ignored during optimisation. We then use the EM

algorithm to minimize the above negative likelihood:

E step. $v_j | Z, Y, a_v, b_v \sim \text{Gamma}\left(\frac{a_v + |N_j|}{2}, \frac{b_v + \sum_{i \in N_j}(z_i - y_{ij})^2}{2}\right).$

M step. Let $q(V) = P(V|Z^{\text{old}}, Y, a_v, b_v)$ and

$$Q(Z; Z^{\text{old}}) = \mathbb{E}_q \log P(Z, Y, V | \mu, \lambda, a_v, b_v) ,$$

then by setting the gradient of the Q function with respect to Z and μ to be zero, we obtain

$$z_i = \frac{\lambda \mu + \sum_{j \in W_i} \mathbb{E}_q v_j y_{ij}}{\lambda + \sum_{j \in W_i} \mathbb{E}_q v_j} , \qquad (4)$$

where

$$\mathbb{E}_q v_j = \frac{a_v + |N_j|}{b_v + \sum_{i \in N_j}(z_i - y_{ij})^2} , \qquad (5)$$

$$\mu = \frac{1}{N} \sum_i z_i .$$

3.1.3 Interpretations.

z_i and its update rule. In Equation (4), z_i is updated by a weighted arithmetic average of μ and $\{y_{ij}\}_{j \in W_i}$, weighted by λ and $\{\mathbb{E}_q v_j\}_{j \in W_i}$, respectively. Higher weights lead to a larger impact on average, so these weights can be considered as encoding worker reliability. In particular, λ is the weight for a "default worker" who always labels every item as μ.

v_j and its estimation rule. In Equation (5), $|N_j|$ is the number of items that worker j has labelled, $\sum_{i \in N_j}(z_i - y_{ij})^2$ is the sum of squared errors (SSE$_j$) that worker j has made. Note that $y_{ij} \in \{0, 1\}$ and $z_i \in [0, 1]$, thus $0 \le \text{SSE}_j \le |N_j|$. In the case that we know the true label, z_i is also binary just like y_{ij}, then SSE$_j$ is simply the number of wrong judgements that worker j has made. The estimation of $\mathbb{E}_q v_j$ can be considered as a smoothed inverse error rate with a_v and b_v serving as the smoothing parameters. The larger the $\mathbb{E}_q v_j$, the smaller the worker j's smoothed error rate is.

Hyper-parameters. The values of a_v and b_v encode our belief that worker j has labelled a_v items before and b_v judgements were wrong, which naturally suggests $b_v \le a_v$ and further ensures $\mathbb{E}_q v_j \ge 1$. So when $b_v \le a_v$, even if all labels from worker j in task k are wrong, we still have $\mathbb{E}_q v_j \ge 1$. This mechanism guarantees everyone an unconditional minimum weight of 1 to vote for the truth. We may consider setting $b_v > a_v$ so that $\mathbb{E}_q v_j$ is allowed to be smaller than 1, however, in this setting, a worker who gets 10/10 wrong will be penalized more than another worker who gets 1000/1000 wrong due to the weaker smoothing effect on the latter. Hence both estimations are smaller than 1, but the estimation for the latter is closer to 1. As this is not the desired effect, we recommend the natural setting of $b_v \le a_v$. As 1 is the lower bound of $\mathbb{E}_q v_j$, we choose $\lambda = 1$ so that the "default worker" has limited ability to compete with other workers, but is most helpful in tie breaking.

3.2 Multi-class Extension

So far, our proposed models assume binary labels. However, it's easy to extend them to the multi-class setting by employing a one-versus-rest strategy, a commonly used reduction to binary classifiers such as SVMs.

Let K denote the number of classes, $k \in \{0, \dots, K\}$ a class index, y_{kij} a binary indicator that equals 1 if worker j labels item i as class k otherwise 0, Y_k the collection of y_{kij}'s. Then our models can work on Y_k which is binary and output Z_k, a collection of z_{ki}'s, where z_{ki} is the score that our model gives to item i. The higher z_{ki} is, the more likely our model believes item i's true label is k. For item i, our model generates a score z_{ki} for each k value given different Y_k. Finally, $k^* = \arg\max_k z_{ki}$ is picked as the estimated true label for item i.

3.3 Hyper-parameter Settings

We discuss how to set b_v when a_v is given in this section. As a_v and b_v translate to our prior belief that every worker has labelled a_v items and made b_v mistakes, $b_v/a_v = \varepsilon$ is our assumed error rate. This error rate should depend on the quality of worker labels in a dataset, so we seek to estimate ε instead of setting it to be constant regardless of datasets. However, estimating ε requires true labels which we don't have, so a natural way is to estimate them by a simple aggregation method such as majority voting, then calculate ε based on the estimates.

We first consider the binary case. Let $\hat{z}_i = \frac{1}{|W_i|} \sum_{j \in W_i} y_{ij}$ to be the majority voting results, then the overall error rate ε is

$$\varepsilon = \frac{\sum_i \sum_{j \in W_i}(y_{ij} - \hat{z}_i)^2}{\sum_i |W_i|} .$$

To simplify ε, we define n_{i0} and n_{i1} to be the number of workers who have labelled item i as 0 or 1, respectively. Then $|W_i| = n_{i0} + n_{i1}$, $\hat{z}_i = \frac{n_{i1}}{n_{i0} + n_{i1}}$ and we obtain

$$\varepsilon = \frac{\sum_i n_{i0}(0 - \hat{z}_i)^2 + n_{i1}(1 - \hat{z}_i)^2}{\sum_i n_{i0} + n_{i1}}$$

$$= \frac{\sum_i n_{i0}\left(\frac{n_{i1}}{n_{i0} + n_{i1}}\right)^2 + n_{i1}\left(\frac{n_{i0}}{n_{i0} + n_{i1}}\right)^2}{\sum_i n_{i0} + n_{i1}} = \frac{\sum_i \frac{n_{i0} n_{i1}}{n_{i0} + n_{i1}}}{\sum_i n_{i0} + n_{i1}} . \quad (6)$$

Then we can set $b_v = a_v \varepsilon$ to encode that every worker has already annotated a_v items, with $a_v \varepsilon$ mistakes, i.e. an error rate ε.

However, we find that the error rate is underestimated by Equation (6) in practice. We then derive that the error rate given by Equation (6) has an upper bound 1/4:

$$\varepsilon = \frac{\sum_i \frac{1}{n_{i0} + n_{i1}} n_{i0} n_{i1}}{\sum_i n_{i0} + n_{i1}}$$

$$\le \frac{\sum_i \frac{1}{n_{i0} + n_{i1}}\left(\frac{n_{i0} + n_{i1}}{2}\right)^2}{\sum_i n_{i0} + n_{i1}} = \frac{\sum_i \frac{n_{i0} + n_{i1}}{4}}{\sum_i n_{i0} + n_{i1}} = \frac{1}{4} .$$

So given $\varepsilon \in [0, 1/4]$, setting $b_v = a_v \varepsilon$ means our prior belief is that every worker has an accuracy of 75% or higher even in the worst case, which is too optimistic and often doesn't reflect reality. We then propose doubling ε to extend its range to $[0, 1/2]$ to encode our belief that every worker is performing slightly better than random guessing.

For multi-class tasks, as there are K classes, we expect random guessing can achieve an accuracy of $1/K$, i.e. an error rate of $1 - \frac{1}{K}$. To encode that the prior belief that every worker is better than random guessing, we extend the range of ε to $[0, 1 - \frac{1}{K}]$ by multiplying ε with $4(1 - \frac{1}{K})$. This is consistent with the doubling strategy for binary tasks, as $4(1 - \frac{1}{K}) = 2$ when $K = 2$.

4 DISCUSSION

The primary motivation of modelling the truth Z as continuous variables is to make the optimization of finding the most likely Z tractable, i.e. $\arg\max_Z P(Z|Y, \alpha)$. We first describe the difference between this objective and other probabilistic models' objectives, then discuss some interesting findings due to the continuous modelling of Z.

4.1 An Objective Intractable for Other Models

Consider a graphical model defining joint $P(Z, Y, V|\alpha)$. It is often easy to sum out Z or integrate out V in $P(Z, Y, V|\alpha)$ to get $P(Y, V|\alpha)$ or $P(Z, Y|\alpha)$. Ideally, the most likely Z is computed by

$$Z^* = \arg\max_Z P(Z|Y, \alpha), \text{ or equivalently } Z^* = \arg\max_Z P(Z, Y|\alpha),$$

because $P(Z, Y|\alpha) \propto P(Z|Y, \alpha)$. However, when Z is discrete, as in most existing probabilistic models for truth inference, the above objective becomes an intractable discrete optimization task. Typical workarounds are:

(1) Solve $V^* = \arg\max_V P(Y, V|\alpha)$ first, then set $V = V^*$ into the joint distribution which naturally decouples z_i's, so $Z^* = \arg\max_Z P(Y, Z, V^*|\alpha)$ becomes tractable as z_i's can be optimized independently.

$$Z^* = \arg\max_Z P(Y, Z, V^*|\alpha)$$

where $V^* = \arg\max_V P(Y, V|\alpha)$. The compromise here is that although V^* is optimal, Z^* is only optimal when $V = V^*$, which does not capture uncertainty. Examples of methods taking this approach include DS [3] and LFC [18].

(2) Bayesian methods naturally use the posterior to make inference. In $P(Z, Y, V|\alpha)$, only Z is the variable of interest, so V is marginalized first, leaving $P(Z, Y|\alpha)$. The posterior of Z, $P(Z|Y, \alpha)$ is proportional to $P(Z, Y|\alpha)$, however, the normalizing constant $P(Y|\alpha)$ is typically intractable to compute. There are two common ways to deal with this intractability.

- *Variational approaches* [8] approximate the posterior $P(Z|Y, \alpha)$ by another distribution $q(Z)$, which is then used to make inference. Mean-field variational inference is very popular and assumes $q(Z)$ is factorized into $\prod_i q(z_i)$, then the true label of item i is estimated as $\arg\max_{z_i} q(z_i)$, which approximates $\arg\max_{z_i} P(z_i|Y, \alpha)$ where $P(z_i|Y, \alpha)$ is the marginal distribution of z_i.
- *Sampling techniques* generate samples from $P(Z|Y, \alpha)$. Markov chain Monte Carlo methods are commonly used to generate samples as they can work without knowing the normalizing constant $P(Y|\alpha)$. Then samples are used to approximate the marginal distribution over z_i, i.e. $P(z_i|Y, \alpha)$. Finally, $\arg\max_{z_i} P(z_i|Y, \alpha)$ is picked as the estimate of item i's true label.

In summary, with either variational approaches or sampling, the goal of Bayesian methods is to solve

$$z_i^* = \arg\max_{z_i} P(z_i|Y, \alpha).$$

However, both approaches face the shortcoming that the variables of interest (z_i) must be optimized independently according to their marginal distributions, as joint optimization is intractable as Z is discrete. Examples include iBCC [10] and CBCC [22].

By contrast, our methods' objectives are ideal:

$$Z^* = \arg\max_Z P(Z|Y, \alpha).$$

The only compromise we make is that we have to model Z as continuous random variables.

4.2 A Novel Perspective on the Wisdom of Crowds

For our proposed BWA model, the negative log likelihood $-\log P(Z, Y|\alpha)$ defined in Equation (3) is reminiscent of a traditional sum-of-square-error objective as minimized in regression models, except that here the error terms are contained in separate log functions. Reorganizing the expression, and using the sum of squared errors defined as $\mathrm{SSE}_j = \sum_{i \in N_j} (z_i - y_{ij})^2$, we obtain

$$-\log P(Z, Y|\mu, \lambda, a_v, b_v)$$

$$= \frac{\lambda}{2} \sum_i (z_i - \mu)^2 + \frac{1}{2} \sum_j \log\left(\frac{b_v + \mathrm{SSE}_j}{a_v + |N_j|}\right)^{a_v + |N_j|} + const.$$

The exponential of the RHS yields the equivalent objective

$$\arg\min_{Z, \mu} -\log P(Z, Y|\mu, \lambda, a_v, b_v)$$

$$= \arg\min_{Z, \mu} \prod_{\text{item } i} e^{\lambda(z_i - \mu)^2} \prod_{\text{worker } j} \left(\frac{b_v + \mathrm{SSE}_j}{a_v + |N_j|}\right)^{a_v + |N_j|}.$$

The first product over items serves as a regularization term for z_i. For μ, it simply means the minimum can be reached when μ is the mean of all z_i. The product over workers is more interesting. Note that $\frac{b_v + \mathrm{SSE}_j}{a_v + |N_j|}$ also appears in $\mathbb{E}_q v_j$'s update rule, and can be considered as the smoothed error rate of worker j. The exponent, $a_v + |N_j|$, is just the smoothed number of items worker j has labelled. Accordingly, this factor measures the likelihood that all workers make all their annotations of the data incorrectly, with a uniform error rate per worker.

Therefore, the underlying principle behind our proposed BWA model is that *it's unlikely that all workers make mistakes for all their tasks*. Our proposed BWA model infers a truth that minimizes the likelihood of everybody performing incorrectly. This principle makes our model different from other models which seek to explain worker labels through maximising the likelihood of realizing these observations. Our principle is conservative, yet intuitively plausible, and a novel perspective on the *wisdom of crowds*.

4.3 A Novel Constraint on Worker Reliability

In the base model, if we were to force all y_{ij} and z_i to be binary variables ($\{0, 1\}$) and then re-normalize the distributions accordingly, we obtain a new link function between y_{ij} and z_i,

$$P(y_{ij} = 1|z_i, v_j) = \frac{\sqrt{\frac{v_j}{2\pi}} \exp\left\{-\frac{v_j}{2}(z_i - 1)^2\right\}}{\sqrt{\frac{v_j}{2\pi}} \exp\left\{-\frac{v_j}{2}z_i^2\right\} + \sqrt{\frac{v_j}{2\pi}} \exp\left\{-\frac{v_j}{2}(z_i - 1)^2\right\}}$$

$$= \frac{1}{\exp\left\{-v_j(z_i - \frac{1}{2})\right\} + 1} \text{ for } z_i \in \{0, 1\},$$

or in an equivalent but more compact form,

$$P(y_{ij}|z_i, v_j) = \frac{\exp\left\{\frac{1}{2}v_j \mathbf{1}[y_{ij} = z_i]\right\}}{1 + \exp\left\{\frac{1}{2}v_j\right\}},$$

which means each worker has an accuracy of $\sqrt{e^{v_j}}/(1 + \sqrt{e^{v_j}})$, irrespective of the true label z_i. Such behaviour can be captured by a symmetric confusion matrix

$$\frac{1}{1 + \sqrt{e^{v_j}}} \begin{bmatrix} \sqrt{e^{v_j}} & 1 \\ 1 & \sqrt{e^{v_j}} \end{bmatrix}.$$

Due to v_j having a Gamma prior, v_j must be non-negative. Consequently the values on the diagonal must be larger than or equal to the values off diagonal. Accordingly our approach is capable of modelling: a perfect worker, $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, a typical worker, $\begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$, and a random worker, $\begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$. However adversarial workers, e.g., $\begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix}$, cannot be represented. This limitation arises from the choice of prior (Gamma), and does not feature in other confusion matrix based models.

This constraint means our models can't capture adversarial workers, however this limitation fits with empirical observations of crowd-workers: in practice adversarial workers are rare, and are largely filtered out based on performance history, which is a consequence of the data acquisition method, but it's possible that less carefully collected data with less rigorous quality control may not be a good fit to our model. Typically in a confusion matrix based probabilistic model, worker labels are assumed to be drawn from categorical distributions parameterized by rows in confusion matrices, and every row in a confusion matrix is assumed to be drawn from a Beta or Dirichlet distribution. For binary tasks, the constraint that values on the diagonal are larger than or equal to the values off the diagonal could be achieved by using a truncated Beta prior with a support $[0.5, 1]$, but at a much higher computational cost because the distributions will no longer be conjugate.

5 EXPERIMENTS

Initialization. Our techniques use the EM algorithm, and thus require initialization of either μ and $\mathbb{E}_q V$, or Z. We initialise Z, the easier of the two, using majority voting which allows for fast convergence. In some cases, from the MV initialization our algorithm doesn't converge to the global optimum (the highest likelihood). A common solution is to run the algorithm with random initialization multiple times and then find the best likelihood (including a majority voting run), but to obtain deterministic results, we only run our algorithm with the MV initialization in all experiments.

Since we use the released implementations of existing methods, we choose to keep their code unchanged. Implementations initializing parameters randomly (CATD, CRH) are run 10 times on every dataset and the mean accuracy of 10 runs is reported as its accuracy on that dataset.

Hyperparameter settings. Following the suggestions described in Section 3.1.3, we set $\lambda = 1$. We choose a_v and b_v in Section 5.1. We set the tolerance in the stopping criteria to 10^{-3}, such that iteration of the algorithm stops when the relative difference between every z_i and its last value is within 0.1%.

Methods. Zheng et al. [28][2] compared 17 existing aggregation methods and released their implementations, and 10 of them supporting the multi-class setting are used in our experiment including MV, ZC [4], GLAD [26], DS [3], Minimax [29], iBCC [10], CBCC [22], LFC [18], CATD [12], and CRH [1].

Datasets. There are 19 real-world datasets used in the experiments originating with four crowdsourcing dataset collections, namely the union of the CrowdScale 2013's shared task challenge [9][3] (2 datasets), the Active Crowd Toolkit project [23][4] (8 datasets), the Truth Inference in Crowdsourcing project [28][5] (7 datasets), and the GitHub repository for SpectralMethodsMeetEM paper [27][6] (5 datasets). Note that 3 datasets are in common between the last two collections. Table 2 presents some statistics about the 19 datasets.

The first two, senti and fact, are from the CrowdScale 2013's shared task challenge, and are the largest two in our experiment, with 569k and 217k labels respectively. The senti data asks the workers to judge the sentiment of a tweet discussing weather and label it within {Negative, Neutral, Positive, Unrelated, Can't tell}. The fact data requests workers to judge whether relational statements about public figures are correct, e.g. "Stephen Hawking graduated from Oxford". Workers may choose from {Yes, No, Skip}. Note that "Can't tell" or "Skip" may also appear as the true labels of some items due to the inherent ambiguity of some questions. None of the methods compared in our experiments explicitly model the unsure options, instead treating these as a standard label in a 5 and 3-class setting, respectively.

The next 8 datasets are all from the Active Crowd Toolkit project. MS, collected by Rodrigues et al. [19], asks workers to listen to short music samples and to classify the music into 10 music genres, {country, jazz, disco, pop, reggae, rock, blues, classical, mental, hip-hop}. ZC_in and ZC_us, provided by Demartini et al. [4], ask workers to judge if a provided Uniform Resource Identifier (URI) is relevant to a named entity extracted from news, where every URI describes an entity. ZC_all is a combination of ZC_in and ZC_us, and all ZC datasets share the same set of items. The two SPs are movie review datasets with binary sentiment labels. CF is just a small part of the full senti. CF* and SP* have the same items as CF and SP, but were reannotated.

The next 5 datasets are from the Truth Inference project Zheng et al. [28]. prod, collected by Wang et al. [24] for entity resolution, asks workers whether two product descriptions refer to the same product or not. tweet, collected by Zheng et al. [28], contains tweets where every tweet is related to a company. Workers are asked to identify if the tweet has positive sentiment towards the company in the tweet. dog contains dog images from ImageNet [11] and the task is to recognize a breed (out of Norfolk Terrier, Norwich Terrier, Irish Wolfhound, and Scottish Deerhound) for a given dog. face, provided by Mozafari et al. [15], contains face images, and workers are asked to judge whether the person in the image is happy, sad, angry, or neutral. adult, released by Mason and Suri [13], asks workers to determine the age appropriateness (P, PG, R, X) of a website given its link.

Table 2: Statistics about 19 real-world datasets used in our experiments.

Dataset	#item(N)	#worker(W)	#class	#label	#truth	#label/N	#label/W
senti	98980	1960	5	569282	1000	5.75	290.45
fact	42624	57	3	214960	576	5.04	3771.23
CF	300	461	5	1720	300	5.73	3.73
CF*	300	110	5	6025	300	20.08	54.77
MS	700	44	10	2945	700	4.21	66.93
SP	4999	203	2	27746	4999	5.55	136.68
SP*	500	143	2	10000	500	20	69.93
ZCall	2040	78	2	20372	2040	9.99	261.18
ZCin	2040	25	2	10626	2040	5.21	425.04
ZCus	2040	74	2	11271	2040	5.53	152.31
prod	8315	176	2	24945	8315	3	141.73
tweet	1000	85	2	20000	1000	20	235.29
dog	807	109	4	8070	807	10	74.04
face	584	27	4	5242	584	8.98	194.15
adult	11040	825	4	89948	333	8.15	109.03
bird	108	39	2	4212	108	39	108
trec	19033	762	2	88385	2275	4.64	115.99
web	2665	177	5	15567	2653	5.84	87.95
rte	800	164	2	8000	800	10	48.78
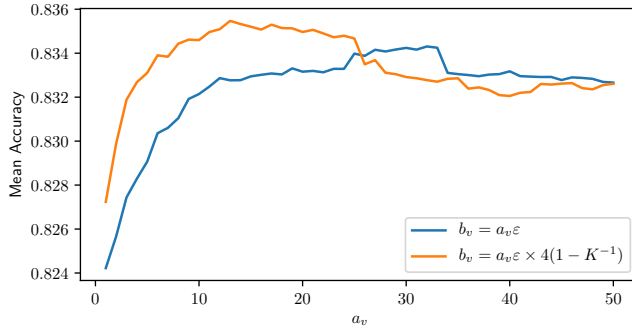
Figure 2: The performance of our proposed BWA model with b_v set by two different strategies.

The last five datasets are used in Zhang et al. [27]. bird, provided by Welinder et al. [25], is a dataset of bird images. The task is to determine whether an image contains at least one duck. The remaining three cover: recognising textual entailment, rte [21], assessing the quality of retrieved documents, trec (TREC 2011 crowdsourcing track)[7], judging the relevance of web search results, web [29].

5.1 Choosing a_v and b_v

Figure 2 shows the mean accuracy of our proposed BWA model across 19 datasets with b_v set by two different strategies described in Section 3.3, namely, using the original error rate ε and using the adjusted error rate $\varepsilon \cdot 4(1 - \frac{1}{K})$. For both strategies, the performance increases steeply when $a_v < 10$ as the prior becomes stronger and provides regularisation to workers who have labelled very few items. We also see both curves are flat around the optimum, so the

[7]https://sites.google.com/site/treccrowd/2011

performance is not very sensitive to the choice of a_v. Due to this reason, we pick a_v from the flat region and report results in both settings: BWA($a_v = 30$, original ε) and BWA($a_v = 15$, adjusted ε).

5.2 Comparison against MV

In this section, we report the results of one-sided Wilcoxon signed rank test performed on every method against MV. We adopt the original version which discards the observations when two methods achieve the same accuracy on a dataset. The null hypothesis H_0 is that the performance difference between a method and MV follows a symmetric distribution around zero.

Table 3 summarizes the results showing whether every method is significantly better than MV and at which level, where significance levels are denoted by asterisks, i.e. 0.1(*), 0.05(**), 0.025(***), respectively. N_r denotes the effective sample size, and W_- the statistic of a method summing up the ranks of datasets that this method is inferior to MV. Zencrowd and iBCC achieve 85.33% accuracy on the CF* dataset which is exactly the same performance as MV resulting, therefore their N_r's are reduced to 18.

The rightmost column in Table 3 shows the p-value calculated based on approximating the cumulative density function (cdf) of W_- given H_0 as a cdf of Gaussian. Due to the small N_r in our experiment (18 or 19), the approximation is somewhat inaccurate, but we keep them there as a reference, as they show ranking of the improvement achieved by all methods comparing to MV.

5.3 Discussion

Scalability. To measure the efficiency of all methods, we run all experiments on a modest desktop with a 7-th generation Intel i5 CPU and 8 GB memory. All methods including our proposed BWA model are implemented in Python, except for iBCC and CBCC

Table 3: One-sided Wilcoxon signed rank test results (all methods against MV)

Method	Mean accuracy	N_r	W_-	Sig. level	Approx. p-value
MV	0.8196				
ZC	0.8294	18	64		0.1746
GLAD	0.8305	19	47	**	0.0267
DS	0.8309	19	71		0.1671
Minimax	0.8141	19	81		0.2866
iBCC	0.8352	18	52	*	0.0723
CBCC	0.8291	19	71		0.1671
LFC	0.8338	19	69		0.1477
CATD	0.8334	19	56	*	0.0583
CRH	0.8054	19	91		0.4361
BWA($a_v = 30$, original ε)	0.8342	18	38	***	0.0193
BWA($a_v = 15$, adjusted ε)	0.8352	18	39	***	0.0214
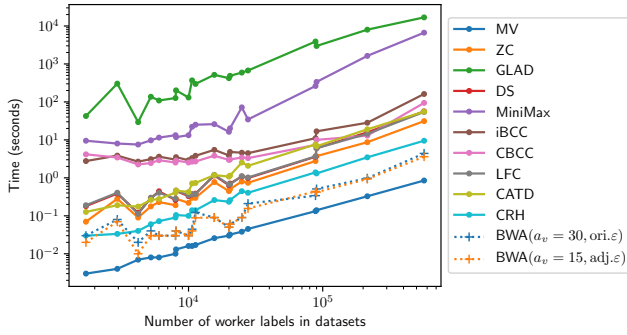
Figure 3: Running time (seconds) of all methods on 19 real-world datasets.

which are implemented in C#, and Minimax is implemented in MATLAB.

As shown in Figure 3, our proposed methods outperform the other existing methods (except MV) in terms of running time. For datasets other than the largest two (senti and fact), our method converged within one second. GLAD and Minimax are the two slowest methods, as they both require gradient-based optimization algorithms, and the required gradients are expensive to compute.

Accuracy. As shown in Table 3, although iBCC has the highest mean accuracy across all datasets, it outperforms MV only at a significance level of 0.1, which is insufficient to reject the null hypothesis. On the other hand, GLAD, whose mean accuracy is not outstanding, significantly outperforms MV at a significance level of 0.05, the best among all 9 benchmarks. In contrast, BWA achieves not only the highest mean accuracy, the same as iBCC, but also strongest significance of outperforming MV, at a level of 0.025.

As shown in Table 3, our proposed BWA($a_v = 15$, adjusted ε) model and iBCC achieve the highest mean accuracy of 83.52%, and our BWA model using another hyper-parameter setting ($a_v = 30$, original ε) which achieves 83.42%, followed by LFC 83.38% and CATD 83.34%. Interestingly, both our proposed methods and CATD model the latent truth z_i's as continuous variables, but in CATD the

optimization objective is heuristic based, while in our models, the optimization objective is derived from explicit probabilistic models. Minimax is an interesting model in that it largely outperforms others on bird and web but performs the worst on MS, the three ZC datasets, and prod. This may be due to Minimax not being a probabilistic model thus its objective function is not well regularised and often too aggressive. This may also explain the results for CRH, another non-probabilistic model.

Figure 4 provides more details of the accuracies of all methods on the 19 real-world datasets. We roughly sort all methods into two categories: confusion-matrix-based methods (DS, MiniMax, iBCC, CBCC, LFC) and 1-coin models (ZC, GLAD, CATD, CRH, our proposed BWA). The name "1-coin" arises from these models only learning a single parameter per worker to capture their accuracy. face, bird, trec are three successful cases for confusion-matrix-based methods, because on those datasets workers behave very differently across classes, which "1-coin" models cannot capture. However, they perform not so well on senti, fact, MS, and ZC datasets due to either the data is overall insufficient to learn a $K \times K$ confusion matrix for every worker, or the class is imbalanced that for some minority classes worker behaviour can't be reliably estimated.

6 CONCLUSION

This paper presents a novel method for aggregating crowdsourced labels for classification. We propose a generative Bayesian model, relaxing a discrete binary problem to a continuous regression task, which we model using a normal likelihood and conjugate inverse-Gamma prior, and can be extended to work in multi-class settings by employing the one-versus-rest strategy.

Our models allow for straightforward inference using expectation-maximisation, which is implemented in 50 lines of code in the appendix. Our models have interpretable parameters and intuitive update rules, which make them easy to understand and use. We also provide detailed practical suggestions such as how to set the hyper-parameters.

Our experimental results show that our proposed method is the only one that significantly outperforms MV at significance level 0.025 and performs competitively to the state-of-the-art method in terms of mean accuracy, across all datasets drawn from several different application areas, with different sizes and differing degrees of label redundancy. Our proposed method also has a total running time on the same datasets second only to MV.

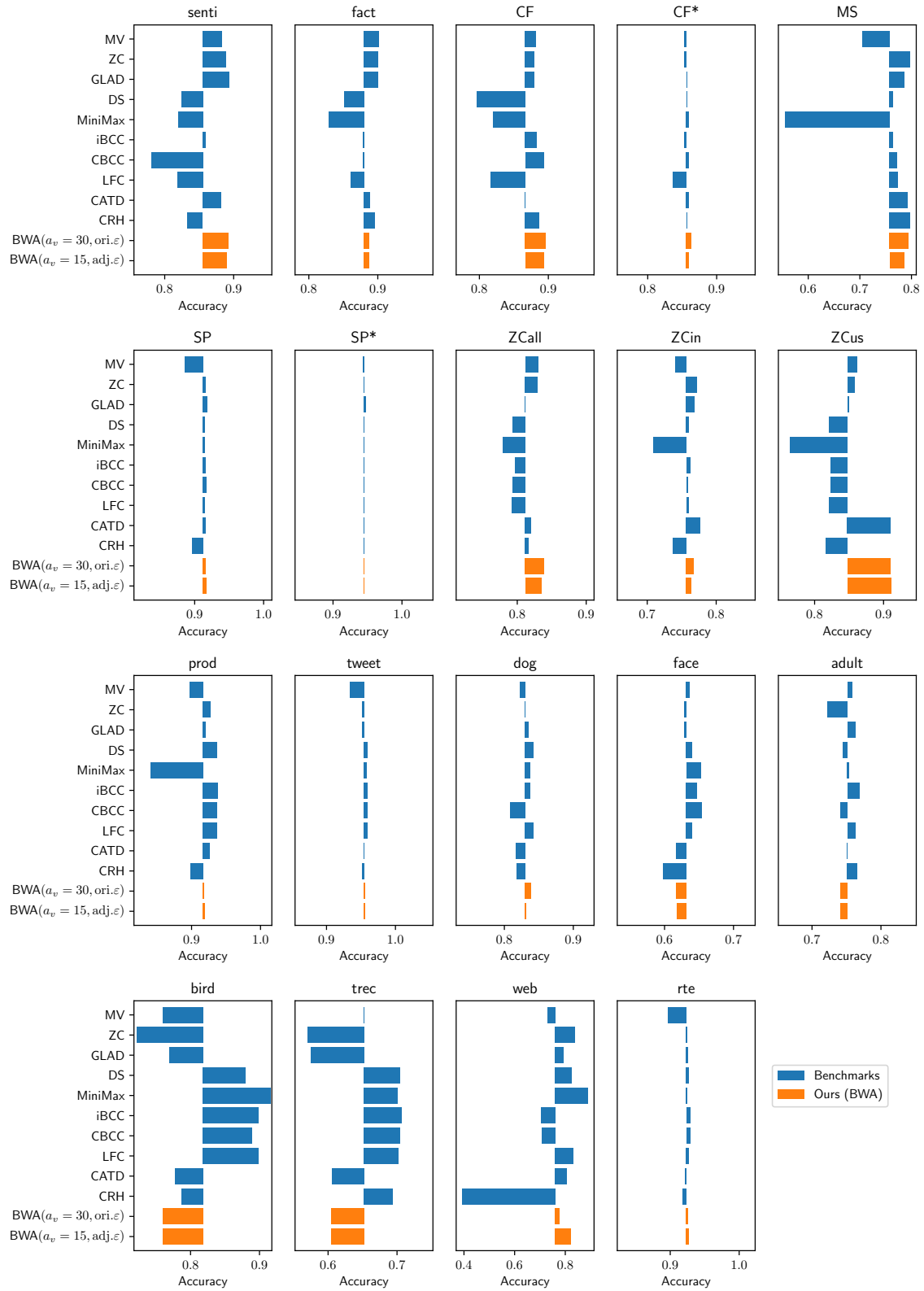Figure 4: Accuracies of all methods on 19 real-world datasets. The horizontal baseline is the mean accuracy across all methods.

A CODE

```python
import numpy as np
import pandas as pd
import scipy.sparse as ssp

def bwa_binary(y_exists_ij, y_is_one_ij, W_i, lambda_, a_v, adj_coef):
    N_j = y_exists_ij.sum(axis=0)
    z_i = y_is_one_ij.sum(axis=-1) / y_exists_ij.sum(axis=-1)

    b_v = a_v * W_i.dot(np.multiply(z_i, 1-z_i)) / y_exists_ij.sum() * adj_coef
    for _ in range(500):
        last_z_i = z_i.copy()

        mu   = z_i.mean()
        v_j = (a_v+N_j) / (b_v + (y_exists_ij.multiply(z_i)-y_is_one_ij).power(2).sum(0))
        z_i = (lambda_*mu + y_is_one_ij.dot(v_j.T)) / (lambda_ + y_exists_ij.dot(v_j.T))

        if np.allclose(last_z_i, z_i, rtol=1e-3): break
    return z_i.A1

def bwa(tuples, a_v=30, lambda_=1, prior_correction=True):
    num_items, num_workers, num_classes = tuples.max(axis=0) + 1
    num_labels = tuples.shape[0]
    W_i = np.bincount(tuples[:, 0])

    adj_coef = 4 * (1 - 1 / num_classes) if prior_correction else 1

    y_exists_ij = ssp.coo_matrix((np.ones(num_labels), tuples[:, :2].T),
                                 shape=(num_items, num_workers), dtype=np.bool).tocsr()
    y_is_one_kij = []
    for k in range(num_classes):
        selected = (tuples[:, 2] == k)
        y_is_one_kij.append(ssp.coo_matrix((np.ones(selected.sum()), tuples[selected, :2].T),
                                           shape=(num_items, num_workers), dtype=np.bool).tocsr())
    z_ik = np.empty((num_items, num_classes))
    for k in range(num_classes):
        z_ik[:, k] = bwa_binary(y_exists_ij, y_is_one_kij[k], W_i, lambda_, a_v, adj_coef)
    return z_ik

def get_acc(predictions, df_truth):
    score = (predictions == predictions.max(axis=1, keepdims=True)).astype(np.float)
    score /= score.sum(axis=1, keepdims=True)
    return score[df_truth.item.values, df_truth.truth.values].sum() / df_truth.shape[0]

data_path = '....'
datasets = ['crowdscale2013/sentiment', 'crowdscale2013/fact_eval']
records = []
for dataset in datasets:
    df_label = pd.read_csv(data_path + dataset + '/label.csv')
    df_label = df_label.drop_duplicates(keep='first')
    prediction_ik = bwa(df_label.values)

    df_truth = pd.read_csv(data_path + dataset + '/truth.csv')
    records.append((dataset, get_acc(prediction_ik, df_truth)))
print(pd.DataFrame.from_records(records, columns=['dataset', 'accuracy']))
## example output
#                       dataset   accuracy
# 0  crowdscale2013/sentiment   0.888000
# 1  crowdscale2013/fact_eval   0.885417

## label.csv example: | truth.csv example:
#                      |
#  item,worker,label   | item,truth
#  26440,1707,3        | 110,0
#  26440,311,3         | 161,1
#  26440,1356,3        | 328,1
#  26440,1774,3        | 611,2
#  ...                 | ...
```

REFERENCES

[1] Bahadir Ismail Aydin, Yavuz Selim Yilmaz, Yaliang Li, Qi Li, Jing Gao, and Murat Demirbas. 2014. Crowdsourcing for Multiple-Choice Question Answering.

[2] Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk.* Association for Computational Linguistics, 1–12.

[3] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* (1979), 20–28.

[4] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web.* ACM, 469–478.

[5] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.

[6] Djellel E. Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. 2012. Mechanical Cheat: Spamming Schemes and Adversarial Techniques on Crowdsourcing Platforms. *CrowdSearch 2012 workshop at WWW* (2012), 26–30.

[7] Jeff Howe. 2008. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business* (1 ed.). Crown Publishing Group, New York, NY, USA.

[8] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. 1999. An introduction to variational methods for graphical models. *Machine learning* 37, 2 (1999), 183–233.

[9] Tatiana Josephy, Matt Lease, Praveen Paritosh, Markus Krause, Mihai Georgescu, Michael Tjalve, and Daniela Braga. 2014. Workshops held at the first AAAI conference on human computation and crowdsourcing: A report. *AI Magazine* 35, 2 (2014), 75–78.

[10] Hyun-Chul Kim and Zoubin Ghahramani. 2012. Bayesian classifier combination. In *Artificial Intelligence and Statistics.* 619–627.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.* 1097–1105.

[12] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. 2014. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment* 8, 4 (2014), 425–436.

[13] Winter Mason and Siddharth Suri. 2012. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior research methods* 44, 1 (2012), 1–23.

[14] Pablo G Moreno, Antonio Artés-Rodríguez, Yee Whye Teh, and Fernando Perez-Cruz. 2015. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research* (2015).

[15] Barzan Mozafari, Purna Sarkar, Michael Franklin, Michael Jordan, and Samuel Madden. 2014. Scaling Up Crowd-sourcing to Very Large Datasets: A Case for Active Learning. *Proc. VLDB Endow.* 8, 2 (Oct. 2014), 125–136. https://doi.org/10.14778/2735471.2735474

[16] Scott Novotney and Chris Callison-Burch. 2010. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 207–215.

[17] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP.* The Association for Computational Linguistics, 2383–2392.

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

[19] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. 2013. Learning from multiple annotators: distinguishing good from random labelers. *Pattern Recognition Letters* 34, 12 (2013), 1428–1436.

[20] Edwin Simpson, Stephen Roberts, Ioannis Psorakis, and Arfon Smith. 2013. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision making and imperfection.* Springer, 1–35.

[21] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and Fast—but is It Good?: Evaluating Non-expert Annotations for Natural Language Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08).* Association for Computational Linguistics, Stroudsburg, PA, USA, 254–263.

[22] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. 2014. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web.* ACM, 155–164.

[23] Matteo Venanzi, Oliver Parson, Alex Rogers, and Nick Jennings. 2015. The ActiveCrowdToolkit: An open-source tool for benchmarking active learning algorithms for crowdsourcing research. In *Third AAAI Conference on Human Computation and Crowdsourcing.*

[24] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. 2012. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment* 5, 11 (2012), 1483–1494.

[25] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. 2010. The multidimensional wisdom of crowds. In *Advances in neural information processing systems.* 2424–2432.

[26] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems.* 2035–2043.

[27] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. 2014. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. In *Advances in neural information processing systems.* 1260–1268.

[28] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.

[29] Denny Zhou, Sumit Basu, Yi Mao, and John C Platt. 2012. Learning from the wisdom of crowds by minimax entropy. In *Advances in neural information processing systems.* 2195–2203.