Exploiting Worker Correlation for Label Aggregation in Crowdsourcing

Yuan Li [1] **Benjamin I. P. Rubinstein** [1] **Trevor Cohn** [1]

Abstract

Crowdsourcing has emerged as a core component of data science pipelines. From collected noisy worker labels, aggregation models that incorporate worker reliability parameters aim to infer a latent true annotation. In this paper, we argue that existing crowdsourcing approaches do not sufficiently model worker correlations observed in practical settings; we propose in response an enhanced Bayesian classifier combination (EBCC) model, with inference based on a mean-field variational approach. An introduced mixture of intra-class reliabilities—connected to tensor decomposition and item clustering—induces inter-worker correlation. EBCC does not suffer the limitations of existing correlation models: intractable marginalisation of missing labels and poor scaling to large worker cohorts. Extensive empirical comparison on 17 real-world datasets sees EBCC achieving the highest mean accuracy across 10 benchmark crowdsourcing methods.

1. Introduction

Production systems for machine learning, natural language processing, computer vision, and information retrieval are regularly trained and evaluated on vast annotated datasets collected by crowdsourcing services (Howe, 2008; Callison-Burch & Dredze, 2010). While crowdsourced annotations are low cost per label, they can be highly noisy, with few annotations available per item, and with labels procured from large cohorts of worker annotators (Difallah et al., 2012). Consequently, inferring consensus aggregation of collected annotations is a core crowdsourcing task, with the simplest technique being majority voting. While simple to implement, majority voting is deficient in granting workers equal votes towards consensus. Numerous probabilistic models have emerged that parameterise worker reliability to improve consensus accuracy (Dawid & Skene, 1979; Whitehill et al., 2009; Kim & Ghahramani, 2012).

In this paper we argue by carefully worked example (Section 4.1) and experimentation on synthetic data (Section 5.1) that modeling correlation between worker labels has significant potential to improve truth inference. We propose a model (Section 4.3) that captures worker correlation by modeling true classes as mixtures of subtypes, with class-level correlation a consequence of worker behaviour varying by subtype. Extending a family of Bayesian classifier combination (BCC) models (Kim & Ghahramani, 2012), we term our model *enhanced BCC* (EBCC) and develop a variational approach for inference (Section 4.4).

While many relevant approaches exist for truth inference (Section 2), very few of them model correlation between workers. Among models that purely rely on crowdsourced labels to infer the truth, the only one incorporating worker correlation, dBCC (Kim & Ghahramani, 2012), has limitations that disqualify it for crowdsourcing (Section 3.1.2): as the missing annotations cannot be tractably marginalised out, all workers must annotate all items; and because dBCC possesses parameters quadratic in the number of workers, it cannot scale to large worker cohorts. (In its original setting of classifier combination, where all classifier predictions are available on only moderately-many classifiers, dBCC's shortcomings are unimportant.) Fortunately our proposed method EBCC suffers no such shortcomings (Section 4.2). We connect our proposed mixture model for classes to tensor decomposition (Sections 4.1, 4.2) and item clustering (Section 4.5), to help explain EBCC's operation.

We conduct extensive experiments on 17 datasets with sources spanning music genre classification, news named entities labeling, movie review and tweet sentiment analysis. Compared to 10 state-of-the-art benchmark methods, EBCC achieves the highest mean accuracy (Section 5).

2. Related Work

Initiating the area of worker label aggregation, Dawid & Skene (1979) used a confusion matrix parameter to generatively model worker labels conditioned on the item's true

[1] School of Computing and Information Systems, University of Melbourne, Victoria, Australia. Correspondence to: Yuan Li <yuanl4@student.unimelb.edu.au>, Benjamin Rubinstein <benjamin.rubinstein@unimelb.edu.au>, Trevor Cohn <trevor.cohn@unimelb.edu.au>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

annotation, for clinical diagnostics. Kim & Ghahramani (2012) formulated a Bayesian generalisation with Dirichlet priors and inference by Gibbs sampling, while Simpson et al. (2013) instead used more efficient variational Bayesian inference. Their analysis of inferred worker confusion matrix clustering is a natural precursor to modelling worker correlation. Taking a parametric hierarchical approach, Venanzi et al. (2014) explicitly modelled workers in clusters within which confusion matrices are likely similar; follow-up work used a non-parametric Dirichlet process for more flexible generation of Dirichlet priors (Moreno et al., 2015). Imamura et al. (2018) derived a minimax error rate for general confusion-matrix-based models and proposed a worker clustering model where the number of clusters can be determined using the derived minimax error rate. Shah et al. (2016) and Khetan & Oh (2016) proposed generalized DS models involving item difficulty for aggregating binary labels and adaptively collecting labels from crowd.

Forgoing confusion matrix parametrisation, Whitehill et al. (2009) proposed an unsupervised item response theory approach which additionally models item difficulty. Another approach, taken by Zhou et al. (2012), estimates true annotations via a minimax entropy principle, which generatively models worker labels from categorical distributions per worker-item pair, promoting distributions close to worker label empirical distributions.

The database community has independently studied truth discovery, merging in entity resolution, and data fusion. Similar to the DS model, but with scalar worker accuracy parameters, the model of Demartini et al. (2012) is used with EM. Li et al. (2014) model worker labels as truth-centred Gaussian noise, with worker ability parametrised by scalar variance. Aydin et al. (2014) iteratively update estimated truth and worker weights so as to minimise the sum of worker weight times distance between worker label and estimated truth.

There is another line of work on jointly learning a classifier and inferring the truth. Cao et al. (2019) used a logistic-regression-style label aggregator considering worker labels as features and making predictions via softmax. Discriminative aggregators are impossible to train using the maximising likelihood principle, but under their proposed MaxMIG framework, such aggregators can be trained jointly with a classifier by maximising their mutual information. In this way, workers producing highly correlated labels can be detected as “redundant features”, so the model is more robust when workers make highly correlated mistakes.

3. Preliminaries

In this section, we first define the crowdsourced annotation aggregation problem and our notation, then discuss two representative Bayesian models for crowdsourcing aggregation.

Notation. Assume there are W workers who classify N items into K categories. Let z_i be the latent true annotation of item i, y_{ij} the label that worker j assigns to item i, $\mathcal{W}_i$ the set of workers who have labelled item i, We use the capitalized letter of a variable to denote the collection of all such variables, for example, Z is $\{z_1, z_2, \ldots, z_N\}$.

3.1. Bayesian Classifier Combination (BCC) Models

The BCC model (Kim & Ghahramani, 2012) was proposed for unsupervised ensembling of discrete outputs from several black-box classifiers. It has been successfully used in crowdsourcing aggregation by making the analogue that workers are black-box classifiers and labels are their discrete outputs (Simpson et al., 2013). The BCC model has several variants. Here we discuss two representative ones, namely independent BCC (iBCC) and dependent BCC (dBCC).

3.1.1. Independent BCC

The iBCC model is a directed graphical model which assumes that given the true label z_i of an item, worker labels to item i are generated independently by different workers,

$$p\left(y_{i1}, \ldots, y_{iW} | z_i\right) = \prod_{j=1}^{W} p(y_{ij}|z_i)\,. \tag{1}$$

We refer to this as the *worker conditional independence assumption*. Furthermore, $p(y_{ij} = l | z_i = k) = v_{jkl}$ is assumed invariant to items. We denote the parameterisation of $p(y_{ij}|z_i = k)$ as $\vec{v}_{jk} = (v_{jk1}, v_{jk2}, \ldots, v_{jkK})$.

An important property of Equation (1) is that in the case that not all workers have labelled item i, the likelihood of its observed labels $\{y_{ij}\}_{j\in\mathcal{W}_i}$ can be calculated easily by marginalising those unobserved labels $\{y_{ij}\}_{j\notin\mathcal{W}_i}$ out,

$$p\left(\{y_{ij}\}_{j\in\mathcal{W}_i} | z_i\right) = \sum_{\{y_{ij}\}_{j\notin\mathcal{W}_i}} p\left(y_{i1}, \ldots, y_{iW} | z_i\right)$$

$$= \sum_{\{y_{ij}\}_{j\notin\mathcal{W}_i}} \prod_{j=1}^{W} p(y_{ij}|z_i) = \prod_{j\in\mathcal{W}_i} p(y_{ij}|z_i)\,. \tag{2}$$

The iBCC model is depicted in Figure 1. Apart from how y_{ij}'s are generated, it assumes that $z_i \sim \text{Categorical}(\vec{\tau})$, $\vec{\tau} \sim \text{Dirichlet}(\vec{\alpha})$, and $\vec{v}_{jk} \sim \text{Dirichlet}(\vec{\beta}_k)$. The joint distribution is

$$p(Y, Z, V, \tau | \alpha, \beta)$$
$$= \prod_i p(z_i|\vec{\tau}) \prod_{j\in\mathcal{W}_i} p(y_{ij}|z_i, V_j) \cdot \text{Dir}(\vec{\tau}|\vec{\alpha}) \prod_k \text{Dir}(\vec{v}_{jk}|\vec{\beta}_k)\,.$$

The iBCC model is a popular extension to the DS model (Dawid & Skene, 1979) and has been independently

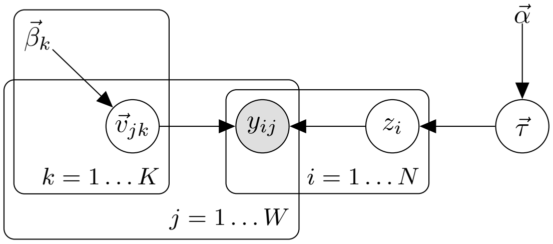

Figure 1. The plate notation for the iBCC model.

re-discovered and implemented using Gibbs sampling (Kim & Ghahramani, 2012; Zhao et al., 2012), mean-field variational Bayes (Simpson et al., 2013; Felt et al., 2015), and expectation propagation (Venanzi et al., 2014). Despite its popularity, underlying independence assumptions prevent the model from capturing correlations between labels from different workers—a serious limitation as we will show.

3.1.2. Dependent BCC

dBCC is an undirected graphical model proposed to overcome the above limitation. In contrast to Equation (1), dBCC uses a Markov network to model the dependence between y_{ij}'s,

$$p\left(y_{i1}, y_{i2}, \ldots, y_{iW} | z_i, V, U\right) = \qquad (3)$$
$$\frac{1}{C(V,U,z_i)} \exp\left\{ \sum_{1 \le j < j' \le W} u_{jj' y_{ij} y_{ij'}} + \sum_{j=1}^{W} v_{j z_i y_{ij}} \right\} ,$$

where $C(V, U, z_i)$ is a partition function that normalises the exponential part; U and V are two matrices of shape (W, W, K, K) and (W, K, K) respectively; $u_{jj'll'}$ relates worker j and j', the larger it is the more likely worker j and j' assign l and l' to the same item; v_{jkl} relates y_{ij} and z_i, the higher it is the more likely worker j labels a class-k item as l. The dBCC model further assumes $u_{jj'll'}$ and v_{jkl} are drawn from Gaussian distributions $\mathcal{N}(0, \sigma_u^2)$ and $\mathcal{N}(0, \sigma_v^2)$ respectively. The generation of z_i is the same as in iBCC.

Note that Equation (3) is the full joint distribution over labels from all workers to item i, but in practice we may only observe labels from a small set of workers, then marginalisation of the full joint is required to calculate the likelihood of observing $\{y_{ij}\}_{j \in \mathcal{W}_i}$,

$$p(\{y_{ij}\}_{j \in \mathcal{W}_i} | z_i, V, U) = \sum_{\{y_{ij}\}_{j \notin \mathcal{W}_i}} p\left(y_{i1}, \ldots, y_{iW} | z_i, V, U\right) .$$

Unfortunately, this marginalisation is intractable owing to the partition function and the full connectivity between y_{ij}'s. Furthermore, even if all workers have labelled all items, thus removing the need of marginalisation, the number of parameters is $O(W^2K^2)$ which is quadratic in W, and as such the model cannot scale to large cohorts of workers.

Table 1. A toy example of two highly correlated workers A and B.

	10 items ($z = 0$)									
worker A	1	1	1	1	1	0	0	0	0	0
worker B	1	1	1	1	0	1	0	0	0	0

The above limitations make dBCC impractical for crowdsourcing aggregation, and motivate our proposed model in the next section.

4. The Proposed Model

4.1. Fitting the Joint Distribution over Worker Labels

We begin with a toy example to illustrate the relation between modelling the correlation between workers and tensor decomposition. Suppose there are two workers A and B who have labelled 10 class-0 items, the labels they generate are shown in Table 1.

The joint distribution over their labels is

$$p(y_A, y_B | z = 0) = \begin{array}{cc} y_A = 0 & y_A = 1 \\ \left[\begin{array}{c} 0.4 \\ 0.1 \end{array}\right. & \left.\begin{array}{c} 0.1 \\ 0.4 \end{array}\right] \end{array} \begin{array}{c} \\ y_B = 0 \\ y_B = 1 \end{array} .$$

For each worker, the marginal distribution is $\left[\begin{smallmatrix} 0.5 \\ 0.5 \end{smallmatrix}\right]$, then following Equation (1), we calculate the outer product of two marginal distributions and obtain a very poor approximation to the joint, $\left[\begin{smallmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{smallmatrix}\right]$. This is partly due to being constrained to rank-1 approximations owing to the worker conditional independence assumption. If using two rank-1 matrices, we could obtain a far better approximation,

$$p(y_A, y_B | z{=}0) \approx \frac{1}{2}\begin{bmatrix} .9 \\ .1 \end{bmatrix} \otimes \begin{bmatrix} .9 \\ .1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} .1 \\ .9 \end{bmatrix} \otimes \begin{bmatrix} .1 \\ .9 \end{bmatrix} = \begin{bmatrix} .41 & .09 \\ .09 & .41 \end{bmatrix} ,$$

where $\otimes$ is the tensor product. In general, the joint worker label distribution of more workers can be approximated by a linear combination of more rank-1 tensors, known also as tensor rank decomposition (Hitchcock, 1927), i.e.,

$$p(y_1, \ldots, y_W | z{=}k) \approx \sum_{m=1}^{M} \pi_{km} \vec{v}_{1km} \otimes \cdots \otimes \vec{v}_{Wkm} . \quad (4)$$

This approach is more flexible than the Markov network in dBCC as the quality of approximation can be controlled by the number of components M instead of being constrained to a fixed capacity. Since the number of parameters is $O(WK^2M)$ which is linear in W instead of quadratic, this approach scales to large cohort of workers unlike dBCC.

4.2. Integrating with Tensor Decomposition

We interpret the tensor decomposition as a mixture model where $\vec{v}_{1km} \otimes \cdots \otimes \vec{v}_{Wkm}$ are the mixture components and

π_{km} the mixture weights, so that we have

$$p(y_1, \dots, y_W | z) = \sum_{m=1}^{M} p(g = m|z) \prod_{j=1}^{W} p(y_j | z, g = m) \ .$$

Here g is an auxiliary latent variable used for indexing mixture components. We treat the M components under class k as its M *subtypes* and use subtypes to explain the correlation between worker labels given class k. For example, in Table 1, the first 4 items and the last 4 items could be two subtypes under class 0: a difficult subtype and an easy subtype. This can explain the fact that both workers misclassify the first 4 items and correctly classify the last 4. The remaining 2 items in the middle can be treated as half-difficult and half-easy, thus could belong to two subtypes with probability $[0.5, 0.5]$. The mixture components capture the worker's different behaviours under different subtypes. In this case, two workers have 10% recall on the difficult items and 90% on the easy ones.

Although worker labels are still assumed to be independently generated under subtypes, the assumption is already weaker than that in the iBCC model, endowing the model with more capacity to capture detailed structures under classes.

Unlike the dBCC model, marginalisation is straight-forward in the mixture model. Given a set of workers $\mathcal{W}$, the likelihood of observing $\{y_j | j \in \mathcal{W}\}$ can be written as

$$p(\{y_j\}_{j\in\mathcal{W}} | z) = \sum_{\{y_j\}_{j\notin\mathcal{W}}} p(y_1, \dots, y_W | z)$$

$$= \sum_{m=1}^{M} p(g = m|z) \prod_{j\in\mathcal{W}} p(y_j | z, g = m) \ . \tag{5}$$

4.3. The Generative Process and Joint Distribution

Based on the iBCC model, we add the mixture weight and components index variables $\vec{\pi}_k$ and g_i, and enlarge $\vec{v}_{jkm}$ M times to capture worker behaviour under different subtypes. There are $K \times M$ subtypes in total, and we assume item i belongs to the g_i-th subtype of class z_i. The proposed model is shown in Figure 2 and its generative process is:

1. for k in $1 \dots K$
   (a) $\vec{\pi}_k | a_\pi \sim \mathrm{Dir}(a_\pi \mathbf{1}_M)$
   (b) for m in $1 \dots M$, for j in $1 \dots W$
      - $\vec{v}_{jkm} | \vec{\beta}_k \sim \mathrm{Dir}(\vec{\beta}_k)$
2. $\vec{\tau} | \vec{\alpha} \sim \mathrm{Dir}(\vec{\alpha})$
3. for i in $1 \dots N$
   (a) $z_i | \vec{\tau} \sim \mathrm{Cat}(\vec{\tau})$
   (b) $g_i | \vec{\pi}_{z_i} \sim \mathrm{Cat}(\vec{\pi}_{z_i})$

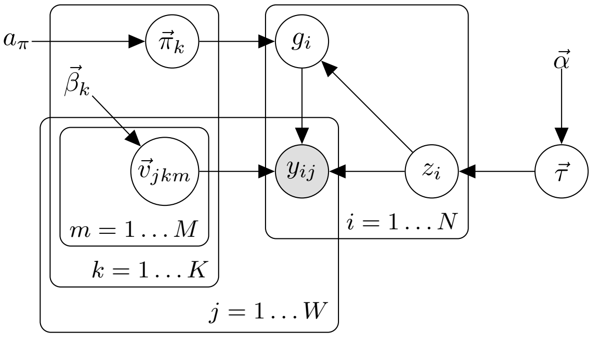

Figure 2. The plate notation for our proposed model.

   (c) for $j \in \mathcal{W}_i$
      - $y_{ij} | \vec{v}_{j z_i g_i} \sim \mathrm{Cat}(\vec{v}_{j z_i g_i})$

The generative process is very similar to that of iBCC's, except that worker reliability is captured at subtype level instead of class level. Following the generative process, the joint distribution is

$$\begin{aligned} & p(\pi, V, \tau, Z, G, Y | a_\pi, \alpha, \beta) \\ =& p(\pi | a_\pi) p(V|\beta) \cdot p(\tau|\alpha) p(Z|\tau) p(G|\pi, Z) p(Y|Z, G, V) \\ \propto & \textstyle\prod_k \prod_m \pi_{km}^{a_\pi - 1} \cdot \prod_j \prod_k \prod_m \prod_l v_{jkml}^{\beta_{kl}-1} \\ & \cdot \textstyle\prod_k \tau_k^{\alpha_k - 1} \cdot \prod_i \tau_{z_i} \cdot \prod_i \pi_{z_i g_i} \cdot \prod_i \prod_{j\in\mathcal{W}_i} v_{j z_i g_i y_{ij}} . \end{aligned}$$

4.4. The Inference Algorithm

The goal of inference is to find the most likely Z given the worker labels Y and all hyper-parameters, i.e. $\arg\max_Z p(Z|Y, a_\pi, \alpha, \beta)$, which is intractable to solve directly. It is possible to derive an expectation maximisation (EM) algorithm for solving $\max_{\pi,V} p(\pi, V | Y, a_\pi, \alpha, \beta)$ so that point estimates for π and V can be obtained, i.e. π^* and V^*, then plug them into $p(Z|\pi^*, V^*, Y, a_\pi, \alpha, \beta)$ to find the most likely Z, just like the EM algorithm for DS.

However, empirical results show that the overall performance of DS is worse than its Bayesian version iBCC (Section 5), due to that the point estimates π^* and V^* lose the uncertainty information. Therefore, we favor a fully Bayesian inference algorithm, and adopt a mean-field variational approach that seeks to find a distribution q that approximates $p(\tau, Z, G, \pi, V | Y, a_\pi, \alpha, \beta)$ so that the following holds

$$\begin{aligned} & \arg\max_Z p(Z|Y, a_\pi, \alpha, \beta) \\ =& \arg\max_Z \textstyle\sum_G \int p(\tau, Z, G, \pi, V | Y, a_\pi, \alpha, \beta) \ \mathrm{d}\tau \mathrm{d}\pi \mathrm{d}V \\ \approx & \arg\max_Z \textstyle\sum_G \int q(\tau, Z, G, \pi, V) \ \mathrm{d}\tau \mathrm{d}\pi \mathrm{d}V \\ =& \arg\max_Z q(Z). \end{aligned}$$

Where q is assumed to be factorised as

$$\begin{aligned} q(\tau, Z, G, \pi, V) =& \mathrm{Dir}(\vec{\tau}|\vec{\nu}) \cdot \textstyle\prod_i q(z_i, g_i) \cdot \prod_k \mathrm{Dir}(\vec{\pi}_k | \vec{\eta}_k) \\ & \cdot \textstyle\prod_k \prod_m \prod_j \mathrm{Dir}(\vec{v}_{kmj} | \vec{\mu}_{kmj}). \end{aligned}$$

Since the joint distribution is fully factorised in q, it's easy to solve $\arg\max_Z q(Z)$ by finding k that maximises every individual $q(z_i = k)$, i.e. $\hat{z}_i = \arg\max_k q(z_i = k)$.

Let $\rho_{ikm} = q(z_i = k, g_i = m)$ and $\gamma_{ik} = q(z_i = k)$, then follow the standard mean-field variational Bayes steps, we can derive the update rules shown below

$$\begin{aligned}
\rho_{ikm} &\propto e^{\mathbb{E}_q \log \tau_k + \mathbb{E}_q \log \pi_{km} + \sum_{j \in \mathcal{W}_i} \mathbb{E}_q \log v_{kmjy_{ij}}} \\
\gamma_{ik} &= \textstyle\sum_m \rho_{ikm} \\
\nu_k &= \alpha_k + \textstyle\sum_i \gamma_{ik} \\
\eta_{km} &= a_\pi + \textstyle\sum_i \rho_{ikm} \\
\mu_{jkml} &= \beta_{kl} + \textstyle\sum_{i \in \mathcal{N}_j} \rho_{ikm} \mathbf{1}[y_{ij} = l] \, .
\end{aligned}$$

The expectations are calculated as follows

$$\begin{aligned}
\mathbb{E}_q \log \tau_k &= \psi(\nu_k) - \psi(\textstyle\sum_k \nu_k) \\
\mathbb{E}_q \log \pi_{km} &= \psi(\eta_{km}) - \psi(\textstyle\sum_m \eta_{km}) \\
\mathbb{E}_q \log v_{jkml} &= \psi(\mu_{jkml}) - \psi(\textstyle\sum_l \mu_{jkml}) \, ,
\end{aligned}$$

where $\psi(\cdot)$ is the digamma function. The Evidence Lower BOund (ELBO) is

$$\begin{aligned}
&\mathbb{E}_q \log p(\tau, Z, G, Y, \pi, V | a_\pi, \alpha, \beta) - \log q(\tau, Z, G, \pi, V) \\
=& \textstyle\sum_k (\nu_k - 1) \mathbb{E}_q \log \tau_k + \sum_k \sum_m (\eta_{km} - 1) \mathbb{E}_q \log \pi_{km} \\
&+ \textstyle\sum_j \sum_k \sum_m \sum_l (\mu_{jkml} - 1) \mathbb{E}_q \log v_{jkml} \\
&- \log B(\vec{\alpha}) - K \log B(a_\pi \mathbf{1}_M) - WM \textstyle\sum_k \log B(\vec{\beta}_k) \\
&+ H(\mathrm{Dir}(\vec{\tau} | \vec{\nu})) + \textstyle\sum_i H(q(z_i, g_i)) + \sum_k H(\mathrm{Dir}(\vec{\pi}_k | \vec{\eta}_k)) \\
&+ \textstyle\sum_j \sum_k \sum_m H(\mathrm{Dir}(\vec{v}_{jkm} | \vec{\mu}_{jkm})),
\end{aligned}$$

where $H(\cdot)$ denotes entropy, and $B(\cdot)$ is the multivariate beta function. The ELBO lower bounds $p(Y | a_\pi, \alpha, \beta)$, so is considered a criterion of how well q approximates p. Because of the factorisation assumption of q, it has one single mode; owing to properties of KL divergence, minimizing $\mathrm{KL}(q \| p)$ will see q approximate one mode in p. However, p is unlikely to have only one mode, therefore one has to run the algorithm many times with different initialisations and pick the best q based on ELBO.

4.5. Comparison to Item Clustering

Both iBCC and our proposed EBCC models can be considered as clustering methods. For an item, although we have no information about its content, its worker labels serve as features. The only unusual thing is that the feature vector of an item may have missing values due to that not all workers have labelled all items. Fortunately, the generative distribution in iBCC and EBCC, as shown in Equation (2) and (5), can handle this by marginalising out missing values.

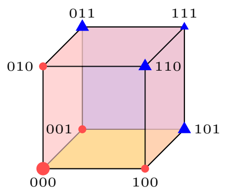

Figure 3. A toy example for item clustering on a cube. ▲ ● denote 300 items and ▴ • 100 items.

Figure 3 shows the distribution over worker labels from 3 workers on 1600 items where all workers have labelled all items. Colors indicate the majority voting aggregation results. Note that this example is symmetric with respect to workers, so all workers are equally good and the majority voting aggregation is reasonable. We run both iBCC and EBCC($M = 3$) on this toy dataset. iBCC fits two clusters with their centroids at 000 and 111,

$$12\% \begin{bmatrix} .99 \\ .01 \end{bmatrix} \otimes \begin{bmatrix} .99 \\ .01 \end{bmatrix} \otimes \begin{bmatrix} .99 \\ .01 \end{bmatrix} + 88\% \begin{bmatrix} .43 \\ .57 \end{bmatrix} \otimes \begin{bmatrix} .43 \\ .57 \end{bmatrix} \otimes \begin{bmatrix} .43 \\ .57 \end{bmatrix} \, .$$

The cluster at 111 is very flat because the most mass of this cluster is at 111's three neighbours and it has to cover them. Consequently, cluster 111 takes some mass from cluster 000 due to its flatness, which makes the latter much sharper. EBCC, on the other hand, is flexible enough to fit all four clusters at 000, 110, 101, and 011,

$$23\% \begin{bmatrix} .9 \\ .1 \end{bmatrix} \otimes \begin{bmatrix} .9 \\ .1 \end{bmatrix} \otimes \begin{bmatrix} .9 \\ .1 \end{bmatrix} + 77\% \left\{ \begin{matrix} \frac{1}{3} \left[\begin{smallmatrix} .9 \\ .1 \end{smallmatrix} \right] \otimes \left[\begin{smallmatrix} .1 \\ .9 \end{smallmatrix} \right] \otimes \left[\begin{smallmatrix} .1 \\ .9 \end{smallmatrix} \right] \\ + \frac{1}{3} \left[\begin{smallmatrix} .1 \\ .9 \end{smallmatrix} \right] \otimes \left[\begin{smallmatrix} .9 \\ .1 \end{smallmatrix} \right] \otimes \left[\begin{smallmatrix} .1 \\ .9 \end{smallmatrix} \right] \\ + \frac{1}{3} \left[\begin{smallmatrix} .1 \\ .9 \end{smallmatrix} \right] \otimes \left[\begin{smallmatrix} .1 \\ .9 \end{smallmatrix} \right] \otimes \left[\begin{smallmatrix} .9 \\ .1 \end{smallmatrix} \right] \end{matrix} \right\} \, ,$$

with three clusters grouped together under class 1. The shapes of four clusters are also similar which is reasonable due to the symmetry of the distribution.

Arguably, iBCC can also fit all four clusters if we relax the constraint that the number of learned clusters has to be the same as the number of classes. We call the relaxed iBCC the Item Clustering model (IC). IC can learn a rectangle $K' \times K$ confusion matrix for every worker, where K' is the number of clusters ($K' > K$). We run IC on the same toy dataset and find it finds the same four clusters as EBCC does, with the portions being $[\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$.

However, IC doesn't model the latent true labels of items, so after obtaining the clustering results, post-processing is required to map K' clusters to K classes. Another drawback is that prior knowledge, such as workers are better than random guessing, can't be encoded in IC because the true label of every cluster is unknown during inference.

Therefore, our proposed EBCC is superior to IC, as the hierarchical generative process of true labels z_i and clusters (subtypes, g_i) defines the mapping from clusters to classes, so that it can directly encode prior knowledge for every class and doesn't require any post-processing.

5. Experiments

Initialisation. We first initialise γ_{ik} by majority voting, i.e. $\gamma_{ik} = \frac{1}{|\mathcal{W}_i|} \sum_{j \in \mathcal{W}_i} \mathbf{1}[y_{ij} = k]$, then multiply it with a random vector drawn from $\mathrm{Dir}(\mathbf{1}_M)$ to initialise $\vec{\rho}_{ik}$. For every dataset, we run the algorithm R times with different random initialisations, and pick the solution with the highest ELBO.

Hyperparameter settings. We set $a_\pi = 0.1/M$ to encourage sparsity of clusters; and $\beta_{kk} = 4$, $\beta_{kk'} = 1, k \neq k'$ to encode that we believe workers are better than random guessing. This is equivalent to assuming that every worker has correctly labelled 4 items under every class, and has made all kinds of mistakes once, i.e. labelling a class-k item as k', $k \neq k'$. We explore two strategies to initialise α_k: (1) set $\alpha_k = 1$ to make the Dirichlet prior for $\vec{\tau}$ uninformative; (2) set $\alpha_k = \sum_i \gamma_{ik}^{(0)}$ where $\gamma_{ik}^{(0)}$ is the MV initialisation for γ_{ik}. The intuition is that MV can provide a reliable estimate of the class portion in the dataset. We use a superscript $^{Emp.}$ to indicate that the second strategy is used.

Number of components. M must be large enough so that the model has capacity to fit the data. Empirically, over all the datasets studied, we find the number of effective subtypes learned by EBCC is < 10 even when larger values of M were used. There are two key reasons: (1) subtypes are learned to explain correlation between workers, which requires pairs of workers to label the same items. Some of the datasets had little overlap between pairs of workers, and thus a small value of M is sufficient; and (2) the Dirichlet prior for $\vec{\pi}_k$, i.e. $\mathrm{Dir}(0.1/M \cdot \mathbf{1}_M)$ used in experiments, encourages sparseness in the distribution of subtypes, therefore the solution tends to use few subtypes. However, if M is too large, the risk of overfitting is increased as the model has much more capacity, also there will be much more local optima as the parameter space increases, so it would be more difficult to converge to the global optimum.

Therefore, we suggest practitioners use $M = 10$ and sufficiently large R, since we observe consistent improvement for all values of M as R increases. We run experiments on real-world datasets with $M = 10, R = 1000$, and synthetic datasets with $M = 2, 5, R = 10$ since the synthetic datasets have much smaller parameter space (only 5 workers) and fewer subtypes per class (2).

5.1. Synthetic Datasets

We run MV, iBCC, and EBCC on synthetic datasets to show that correlations between worker labels can be captured and exploited to assist truth inference. In all datasets, there are 5 workers classifying items into two categories. Every class has two subtypes and all subtypes are distributed evenly with exactly 25% items belonging to each. All worker labels are randomly generated according to their reliability. The first two workers' performances vary on different subtypes with an average accuracy of 50% per class, while the last three workers perform consistently with an accuracy of 70% across subtypes. Table 2 summarises the settings.

Table 2. Accuracy of 5 workers on different subtypes.

z	g	w1	w2	w3	w4	w5	portion
0	0	0.9	0.9	0.7	0.7	0.7	25%
0	1	0.1	0.1	0.7	0.7	0.7	25%
1	0	0.9	0.1	0.7	0.7	0.7	25%
1	1	0.1	0.9	0.7	0.7	0.7	25%

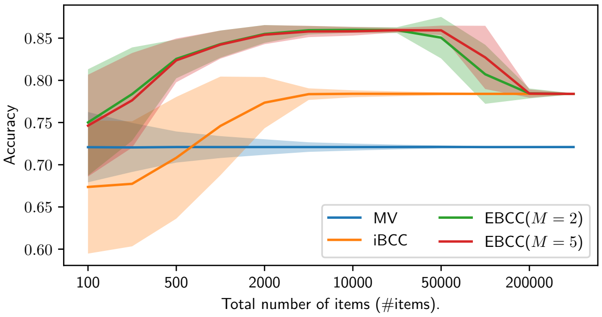

Figure 4. MV, iBCC, and EBCC on synthetic datasets.

The first two workers are likely to agree with each other on class-0 items, but generate different labels to class-1 items. This observation is helpful for inferring the truth, and we expect the EBCC model which directly captures such information will benefit from it. Following the settings in Table 2, we run MV, iBCC, and EBCC ($M = 2, 5$) on datasets with different sizes $\{1, 2, 5\} \times \{10^2, 10^3, 10^4, 10^5\}$. Figures 4 shows the results of three methods. Solid lines show the mean accuracy of 10000 runs for sizes ≤ 5000 and 1000 runs for the remaining, and shaded regions plot mean accuracy $\pm$ one standard deviation of the accuracy.

The performance of MV is very stable at 72.0% while iBCC starts with 67.4% then surpasses MV when $\#\text{items} > 500$ and finally converges to 78.4%. The first two workers are completely random on the class level with their confusion matrices being $[\begin{smallmatrix}0.5 & 0.5\\ 0.5 & 0.5\end{smallmatrix}]$. iBCC can estimate their confusion matrices reliably as more data is available and effectively ignore them during the inference. That's why iBCC converges to the theoretic MV performance of the last three workers: $0.7^3 + 3 \cdot 0.7^2 \cdot 0.3 = 0.784$. EBCC has the ability to capture the special correlation between the first two workers, therefore achieve the highest accuracy 85.9% and consistently outperform MV and iBCC. There is little difference between $M = 2$ and $M = 5$ for EBCC, which suggests insensitivity to over-parameterisation.

Surprisingly, the performance of both EBCC models gets worse when $\#\text{items} > 50\text{k}$ with their variance increasing and decreasing when $50\text{k} \leq \#\text{items} \leq 200\text{k}$. We have examined the estimates of parameters and found that EBCC actually converges to the same solution as iBCC does. But the ELBO of iBCC's solution is lower than the ELBO of the correct solution, therefore we conclude that this is an optimisation problem and that our algorithm gets stuck on bad local optima. As is commonly known, this is a weakness for batch learning on large-scale datasets, therefore we believe a stochastic optimisation algorithm would likely mitigate the problem.

5.2. Real-world Datasets

There are 17 real-world datasets used in this paper coming from three crowdsourcing dataset collections, namely the union of (Venanzi et al., 2015)[1] (8 datasets), (Zheng et al., 2017)[2] (7 datasets), and the GitHub repository for SpectralMethodsMeetEM paper (Zhang et al., 2014)[3] (5 datasets), noting that 3 datasets are in common between the last two collections.

They cover a range of tasks: sentiment analysis for tweets about weather `CF` (Josephy et al., 2014), music genre classification based on 30 sec music samples `MS` and sentiment analysis for movie reviews `SP` (Rodrigues et al., 2013), judging if a provided Uniform Resource Identifier (URI) is relevant to a named entity extracted from news where every URI describes an entity $\texttt{ZC}^{\texttt{all}}$, $\texttt{ZC}^{\texttt{in}}$, $\texttt{ZC}^{\texttt{us}}$ (Demartini et al., 2012), judging whether two product descriptions refer to the same product for entity resolution `prod` (Wang et al., 2012), sentiment analysis for company mentioned in tweets `senti` (Zheng et al., 2017), facial expression classification `face` (Mozafari et al., 2014), judging age-appropriateness (P, PG, R, X) of a website given its link `adult` (Mason & Suri, 2012), determining whether an image contains at least one duck `bird` (Welinder et al., 2010), labeling the breed of dogs `dog` (Zhang et al., 2014), recognising textual entailment `rte` (Snow et al., 2008), assessing the quality of retrieved documents `trec` (TREC 2011 crowdsourcing track)[4], judging the relevance of web search results `web` (Zhou et al., 2012). Finally, $\texttt{CF}^{\star}$ and $\texttt{SP}^{\star}$ are re-annotaed versions of `CF` and `SP` by Venanzi et al. (2015).

Methods. Zheng et al. (2017) compared 17 existing aggregation methods and released their implementations, and 10 of them supporting a multi-class setting are used in our experiment including MV, ZenCrowd (Demartini et al., 2012), GLAD (Whitehill et al., 2009), DS (Dawid & Skene, 1979), Minimax (Zhou et al., 2012), iBCC-EP (Kim & Ghahramani, 2012), CBCC (Venanzi et al., 2014), LFC (Raykar et al., 2010), CATD (Li et al., 2014), and CRH (Aydin et al., 2014). We also include a mean-field variational inference implementation of iBCC (iBCC-MF) to compare with our proposed method implemented by the same technique. We run EBCC with $M = 10, R = 1000$ and two settings for α_k as discussed in hyperparameter settings.

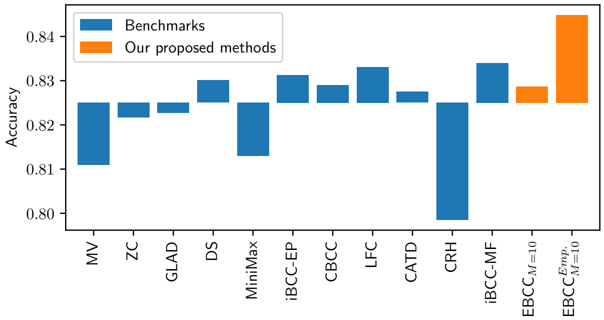

Figure 5. Mean accuracy on 17 real-world datasets. The baseline of the bar plot is the average of all methods' mean accuracies.

Results. Figure 5 shows the mean accuracy of every method on 17 datasets and Figure 6 presents the accuracies on all datasets. More details are provided on our GitHub repository.[5]

$\text{EBCC}^{Emp.}_{M=10}$ has the highest mean accuracy of 84.5%, outperforming the best existing method iBCC-MF which achieves 83.4%. Overall, confusion-matrix-based probabilistic models (EM, iBCC, CBCC, LFC) perform similarly with mean accuracy within range $[82.9\%, 83.4\%]$, followed by three "1-coin" models, namely, CATD (82.8%), GLAD (82.3%), ZC (82.2%). This name arises from these models only learning a single parameter per worker to capture their accuracy. However, a worker may behave differently across classes or subtypes, which "1-coin" models cannot capture. Minimax is an interesting model in that it largely outperforms others on `bird` and `web` but performs the worst on `MS`, the three `ZC`s, 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.

On most datasets, $\text{EBCC}^{Emp.}_{M=10}$ is either the best method, or very close to the best. There are only 4 datasets where EBCC is more than 1% below the best: `web`(12.0%), `adult`(2.11%), `bird`(1.86%), `face`(1.03%). These are the datasets with the lowest average worker accuracy among all datasets: web(37%), face(60%), bird(64%), adult(65%). Note that on `bird` $\text{EBCC}^{Emp.}_{M=10}$ is second only to Minimax, whose unstable performance would prevent it from being used in practice. For the other three datasets, it appears that $\text{EBCC}^{Emp.}_{M=10}$ is overfitting the noise more so than other methods, which follows as EBCC has many more parameters. We suggest using lower capacity models for very noisy

[1] https://github.com/orchidproject/active-crowd-toolkit

[2] https://zhydhkcws.github.io/crowd_truth_inference/index.html

[3] https://github.com/zhangyuc/SpectralMethodsMeetEM

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

[5] https://github.com/yuan-li/truth-inference-at-scale

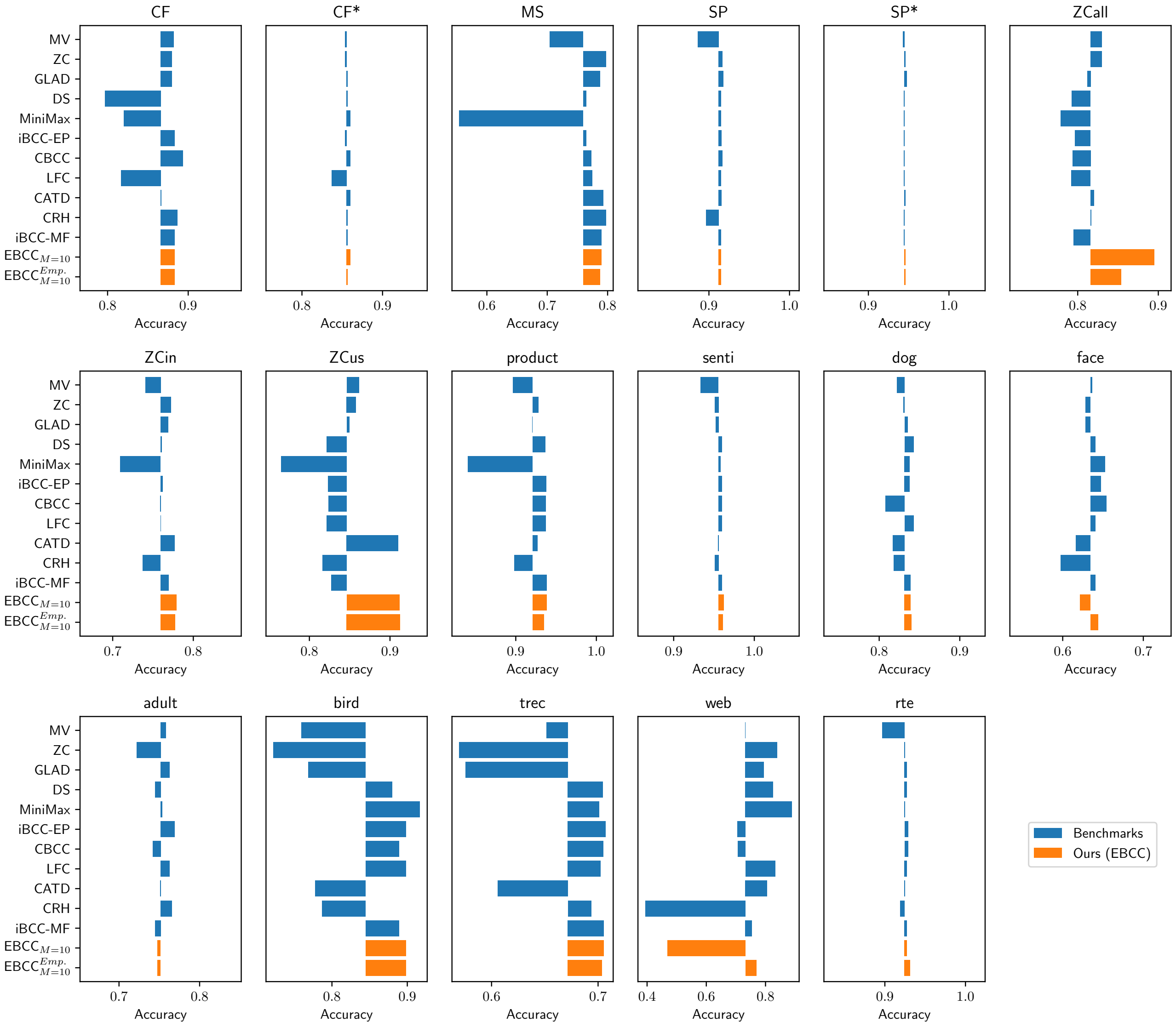

Figure 6. Accuracies of all methods on 17 real-world datasets.

datasets, e.g. iBCC or even 1-coin models such as ZC.

Apart from mean accuracy, we identify a failure case for $\text{EBCC}_{M=10}$. Its accuray on the `web` dataset is 46.9%, which is much lower than $\text{EBCC}^{Emp.}_{M=10}$'s on the same dataset. After examining the estimates of parameters, we found that it learns a very skewed class distribution $\vec{\tau}$ with the value of one class (A) being zero, consequently it doesn't classify any items into class A. Further analysis shows that many workers confuse class A with another two classes, which absorb all clusters from class A in the learning process. Our solution is to set the prior of $\vec{\tau}$ to be the class distribution estimated by majority voting to encode our belief. As shown in Figure 5 and 6, $\text{EBCC}^{Emp.}_{M=10}$ achieves better overall performance than $\text{EBCC}_{M=10}$ does.

6. Conclusion

We have developed a Bayesian model for aggregating crowd-sourced labels that is capable of capturing correlations between labels of different workers. Our model, enhanced Bayesian classifier combination (EBCC), achieves this by introducing a mixture of subtypes per true class, while worker performance varying per subtype induces inter-worker correlation. The efficacy of EBCC is demonstrated in extensive experiments on synthetic data, which confirms the importance of worker correlation, and over a suite of 17 crowd-sourced datasets drawn from a wide variety of domains, where EBCC achieves state-of-the-art for 10/17 of the datasets. We intend to explore the application of stochastic optimisation to EBCC in future work, which should improve the method's robustness.

Acknowledgement

This work was sponsored by Facebook AI and the Defense Advanced Research Projects Agency Information Innovation Office (I2O) under the Low Resource Languages for Emergent Incidents (LORELEI) program issued by DARPA/I2O under Contract No. HR0011-15-C-0114. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. Benjamin Rubinstien was supported by the Australian Research Council, DP150103710.

References

Aydin, B. I., Yilmaz, Y. S., Li, Y., Li, Q., Gao, J., and Demirbas, M. Crowdsourcing for multiple-choice question answering. In *AAAI*, pp. 2946–2953, 2014.

Callison-Burch, C. and Dredze, M. 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*, pp. 1–12. Association for Computational Linguistics, 2010.

Cao, P., Xu, Y., Kong, Y., and Wang, Y. Max-MIG: an information theoretic approach for joint learning from crowds. In *International Conference on Learning Representations*, 2019.

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

Demartini, G., Difallah, D. E., and Cudré-Mauroux, P. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*, pp. 469–478. ACM, 2012.

Difallah, D. E., Demartini, G., and Cudré-Mauroux, P. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. *CrowdSearch 2012 workshop at WWW*, pp. 26–30, 2012.

Felt, P., Black, K., Ringger, E., Seppi, K., and Haertel, R. Early gains matter: A case for preferring generative over discriminative crowdsourcing models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 882–891, 2015.

Hitchcock, F. L. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

Howe, J. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Publishing Group, New York, NY, USA, 1 edition, 2008. ISBN 0307396207, 9780307396204.

Imamura, H., Sato, I., and Sugiyama, M. Analysis of minimax error rate for crowdsourcing and its application to worker clustering model. In *International Conference on Machine Learning*, pp. 2152–2161, 2018.

Josephy, T., Lease, M., Paritosh, P., Krause, M., Georgescu, M., Tjalve, M., and Braga, D. Workshops held at the first aaai conference on human computation and crowdsourcing: A report. *AI Magazine*, 35(2):75–78, 2014.

Khetan, A. and Oh, S. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Advances in Neural Information Processing Systems*, pp. 4844–4852, 2016.

Kim, H.-C. and Ghahramani, Z. Bayesian classifier combination. In *Artificial Intelligence and Statistics*, pp. 619–627, 2012.

Li, Q., Li, Y., Gao, J., Su, L., Zhao, B., Demirbas, M., Fan, W., and Han, J. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment*, 8(4):425–436, 2014.

Mason, W. and Suri, S. Conducting behavioral research on amazon's mechanical turk. *Behavior research methods*, 44(1):1–23, 2012.

Moreno, P. G., Artés-Rodríguez, A., Teh, Y. W., and Perez-Cruz, F. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research*, 2015.

Mozafari, B., Sarkar, P., Franklin, M., Jordan, M., and Madden, S. Scaling up crowd-sourcing to very large datasets: A case for active learning. *Proc. VLDB Endow.*, 8(2):125–136, October 2014. ISSN 2150-8097. doi: 10.14778/2735471.2735474.

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

Rodrigues, F., Pereira, F., and Ribeiro, B. Learning from multiple annotators: distinguishing good from random labelers. *Pattern Recognition Letters*, 34(12):1428–1436, 2013.

Shah, N. B., Balakrishnan, S., and Wainwright, M. J. A permutation-based model for crowd labeling: Optimal estimation and robustness. *arXiv preprint arXiv:1606.09632*, 2016.

Simpson, E., Roberts, S., Psorakis, I., and Smith, A. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision making and imperfection*, pp. 1–35. Springer, 2013.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. 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, pp. 254–263, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

Venanzi, M., Guiver, J., Kazai, G., Kohli, P., and Shokouhi, M. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pp. 155–164. ACM, 2014.

Venanzi, M., Parson, O., Rogers, A., and Jennings, N. The activecrowdtoolkit: An open-source tool for benchmarking active learning algorithms for crowdsourcing research. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

Wang, J., Kraska, T., Franklin, M. J., and Feng, J. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494, 2012.

Welinder, P., Branson, S., Perona, P., and Belongie, S. J. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, pp. 2424–2432, 2010.

Whitehill, J., Wu, T.-f., Bergsma, J., Movellan, J. R., and Ruvolo, P. L. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pp. 2035–2043, 2009.

Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in neural information processing systems*, pp. 1260–1268, 2014.

Zhao, B., Rubinstein, B. I., Gemmell, J., and Han, J. A bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment*, 5(6):550–561, 2012.

Zheng, Y., Li, G., Li, Y., Shan, C., and Cheng, R. Truth inference in crowdsourcing: is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, 2017.

Zhou, D., Basu, S., Mao, Y., and Platt, J. C. Learning from the wisdom of crowds by minimax entropy. In *Advances in neural information processing systems*, pp. 2195–2203, 2012.