Lean Multiclass Crowdsourcing

Grant Van Horn
Caltech

Steve Branson
Caltech

Scott Loarie
iNaturalist

Serge Belongie
Cornell Tech

Pietro Perona
Caltech

Abstract

We introduce a method for efficiently crowdsourcing multiclass annotations in challenging, real world image datasets. Our method is designed to minimize the number of human annotations that are necessary to achieve a desired level of confidence on class labels. It is based on combining models of worker behavior with computer vision. Our method is general: it can handle a large number of classes, worker labels that come from a taxonomy rather than a flat list, and can model the dependence of labels when workers can see a history of previous annotations. Our method may be used as a drop-in replacement for the majority vote algorithms used in online crowdsourcing services that aggregate multiple human annotations into a final consolidated label. In experiments conducted on two real-life applications we find that our method can reduce the number of required annotations by as much as a factor of 5.4 and can reduce the residual annotation error by up to 90% when compared with majority voting. Furthermore, the online risk estimates of the models may be used to sort the annotated collection and minimize subsequent expert review effort.

1. Introduction

Multiclass crowdsourcing is emerging as an important technique in science and industry. For example, a growing number of websites support sharing observations (photographs) of specimens from the natural world and facilitate collaborative, community-driven identification of those observations. Websites such as iNaturalist, eBird, Mushroom Observer, HerpMapper, and LepSnap accumulate large collections of images and identifications, often using majority voting to produce the final species label. Ultimately, this information is aggregated into datasets (*e.g.* GBIF [33]) that enable global biodiversity studies [29]. Thus, the label accuracy of these datasets can have a direct impact on science, conservation and policy. Thanks to the recent dramatic improvements in our field [16, 8, 30, 9], observations collected by these websites can be used to train classification services (*e.g.* see merlin.allaboutbirds.org and inaturalist.org), helping novices label their observations. The result is an even larger collection of observations, but with potentially noisier labels as the number of people taking photos and submitting observations far outpaces the speed at which experts can verify them. The benefits of a simple algorithm like majority vote are lost when the skill of the people contributing labels is uncertain. Thus, there is need for improved meth-

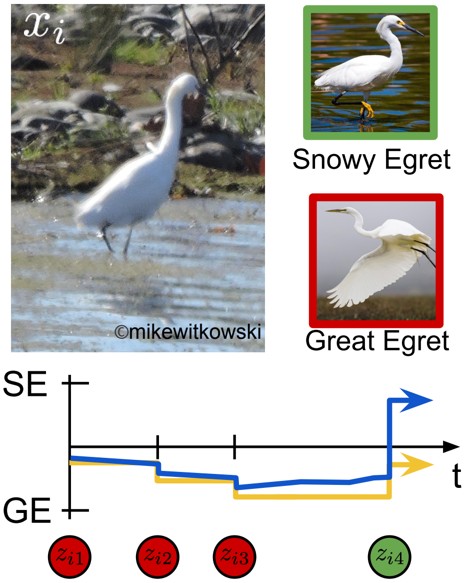


Figure 1: **iNaturalist Community Identification.** A user uploads image x_i (top-left) with an initial species prediction z_{i1} = Great Egret (GE), one out of 1.5k North American bird species. Later, two additional users (potentially alerted that a GE has been spotted) come along and, after inspecting the image *and* the previous identifications, contribute their subjective identifications of the bird species z_{i2} = GE and z_{i3} = GE, agreeing with the uploader. Finally, a fourth user provides a different identification z_{i4} = Snowy Egret (SE). In the plot below the images, two models (red, green) integrate the information differently, with the y axis representing likelihood of SE vs. GE. Majority voting (yellow arrow) simply tallies the vote, and GE is the chosen answer after four votes. Our model (blue arrow) continuously analyzes the users' skills across *other* observations and is therefore capable of updating the likelihood of the predicted label much more frequently. Knowing that the fourth user is highly skilled on these taxa, our model overrides previous users and predicts SE. The underlying ground truth answer is indeed SE. In this work we design and compare several models that estimate user skill and use it to weigh votes appropriately. (View on iNat: https://www.inaturalist.org/observations/4599411)

ods to integrate multiple identifications into a final label.

Figure 1 shows a real example of a user's observation on iNaturalist, a sequence of identifications from the community, and how the current species label is computed using majority voting. The structure of these interactions present three challenges that have not been tackled by prior work on combining multiclass annotations [41, 13, 35, 42, 40, 32, 39]. (1) iNaturalist has a tree structured taxonomy of labels rather than a flat list, allowing users to provide labels at varying depths of the taxonomy depending on their confidence. (2) Identifiers get to see the history of previous identifications for an observation, so their identification is *not* independent of previous identifiers. (3) The number of species under consideration is huge, currently at $\sim$130k but potentially reaching 8M [22].

We propose a new method for aggregating multiple multiclass labels. Our method is based on models of worker behavior and can replace majority vote in websites like iNaturalist, and in more traditional data labeling services (*e.g.* Amazon Mechanical Turk). We show that our models are more accurate than majority voting (reducing error by 90% on data from iNaturalist) and when combined with a computer vision system can drastically reduce the number of labels required per image (*e.g.* by a factor of 5.4 on crowdsourced data). Our main contribution is a method for *multiclass annotation* tasks that (1) can be used in online crowdsourcing, (2) can handle large numbers of classes, (3) can handle a taxonomy of labels allowing workers to respond at coarser levels than leaf nodes, (4) can handle mutually dependent worker labels.

2. Related Work

Kovashka *et al.* [15] provide a thorough review of crowdsourcing techniques for computer vision. The Dawid-Skene (DS) model [5] is the standard probabilistic model for multiclass label inference from multiple annotations. That model assumes each worker has a latent confusion matrix that captures the probability of annotating a class correctly (the diagonal entries) and the probability of confusing two classes (the off diagonal entries). The DS model iteratively infers the reliability of each worker and updates the belief of the true labels, using Expectation-Maximization as the inference algorithm. Alternate inference algorithms for the DS model are based on spectral methods [7, 4, 12, 13, 14, 40], belief propagation [20, 23], expectation maximization [20, 40], maximum entropy [41, 42], weighted majority voting [19, 17] and max-margin [32]. Alternatives to the DS model have also been proposed [28, 10, 37, 36, 26, 31, 11, 39, 2, 3]. Further work based on active learning tackles noisy labelers [21], and task allocation to minimize the monetary cost of dataset construction [13, 14, 27].

Multiclass tasks, as opposed to binary tasks, are explored by [41, 13, 35, 42, 40, 32, 39, 3]. Zhou *et al.* use entropy maximization to model both worker confusions and task difficulties for multiclass [41] and ordinal [42] data. Similarly, Chen *et al.* [3] use max-margin techniques to further improve results for ordinal tasks. Karger *et al.* [13] use an iterative algorithm by converting k-class tasks into $k-1$ binary tasks but makes assumptions on the number of items and workers. Vempaty *et al.* [35] also convert k-class tasks into binary tasks, but take a coding theoretic approach to estimate labels. Zhang *et al.* [40] use spectral methods to initialize the EM inference algorithm of the Dawid-Skene model, while Tian *et al.* [32] fuse a max-margin estimator and the Dawid-Skene model. Zhang *et al.* [39] create probabilistic features for each item and use a clustering algorithm to assign them their final labels, however they do not produce an estimate of worker skill. All of the previous approaches assume that annotations are independent. We differentiate our work by handling both independent *and* dependent annotations collected by sites like iNaturalist. Furthermore, we explore the challenges of "large-scale" multiclass task modeling where the number of classes is nearly $10\times$ larger than the prior art has explored. Our work also handles taxonomic modeling of the classes and non-leaf node worker annotations. See Table 2 for a performance comparison of our model to prior art.

Final label quality between independent and dependent crowdsourcing tasks is studied by Little *et al.* [18], but without modeling workers. The work of Branson *et al.* [2] is the closest to ours, as we adapt their framework to multiclass annotation, which they did not investigate. Furthermore, we explore taxonomic multiclass annotations to reduce the number of parameters. Additionally, we develop models that do not depend on the assumption that worker annotations are independent, and we are thus able to handle mutually dependent annotations where each worker can see previous labels.

3. Multiclass Online Crowdsourcing

Given a set of worker annotations Z for a dataset of images X, the probabilistic framework of Branson *et al.* [2] jointly models worker skill W, image difficultly D, ground truth labels Y, and computer vision system parameters θ. A tiered prior system is used to make the system more robust by regularizing the per worker skill and image difficulty priors. Alternating maximization is used for parameter estimation. The Bayesian risk $\mathcal{R}(\bar{y}_i)$ (see Eq.1 from [2]) can be computed for each predicted label, providing an intuitive online stopping criteria (*i.e.* the model can "retire" images as soon as their risk is below a threshold τ_ϵ). In this work, we extend this framework by implementing multiple models of worker skill for the task of multiclass annotation for independent and dependent worker labels. For our experiments we removed the image difficulty part of the framework and focused solely on modeling workers and their la-

Name	Interpretation	Model	Expression	# Params	# Params For Birds
Flat Single Binomial	Probability of being correct is the same for all species	$z = y$ is binomial with the same parameters regardless of y	$p(z\mid y) = \begin{cases} m & \text{if } z = y \\ (1-m)p(z) & \text{otherwise} \end{cases}$	1	1
Flat Per Class Binomial	Probability of being correct for each species separately	For each value $y = c$, $z = y$ is binomial	$p(z\mid y) = \begin{cases} M(y) & \text{if } z = y \\ (1-M(y))p(z) & \text{otherwise} \end{cases}$	C	1,572
Flat Per Class Multinomial	Confusion probability over each pair of species	For each value $y = c$, z is multinomial	$p(z\mid y) = M(y,z)$	C^2	2,471,184
Taxonomic Single Binomial	Probability of being correct is the same for each species in a genus	$z^l = y^l \mid z^{l-1} = y^{l-1}$ is binomial with the same parameters regardless of y^l	$p(z\mid y) = \prod_l p(z^l\mid y^l)$ $p(z^l\mid y^l) = \begin{cases} m_{y^{l-1}} & \text{if } z^l = y^l \\ (1-m_{y^{l-1}})p(z^l) & \text{otherwise} \end{cases}$	$\|N\| - C$	383
Taxonomic Per Class Binomial	Probability of being correct for each species separately	For each value $y^l = c$, $z^l = y^l \mid z^{l-1} = y^{l-1}$ is binomial	$p(z\mid y) = \prod_l p(z^l\mid y^l)$ $p(z^l\mid y^l) = \begin{cases} M_{y^{l-1}}(y^l) & \text{if } z^l = y^l \\ (1-M_{y^{l-1}}(y^l))p(z) & \text{otherwise} \end{cases}$	$\|N\|$	1955
Taxonomic Per Class Multinomial	Confusion probability for each pair of species in a genus	For each value $y^l = c$, $z^l \mid z^{l-1} = y^{l-1}$ is multinomial	$p(z\mid y) = \prod_l p(z^l\mid y^l)$ $p(z^l\mid y^l) = M_{y^{l-1}}(y^l, z^l)$	$\sum_{n \in N} \lvert\text{children}(n)\rvert^2$	22,472

Table 1: Different options for modeling worker skill given a taxonomy of classes. N is the set of nodes in the taxonomic tree, C is the number of leaf nodes (*i.e.* class labels). The last column shows the number of resulting parameters when modeling the 1,572 species of North American birds and their taxonomy from the iNaturalist database, *for a single worker*. Multinomial models have significantly more parameters but can model commonly confused classes. Taxonomic methods have the benefit of supporting non-species-level human responses, modeling skill at certain taxa, and reducing the number of parameters for multinomial models.

bels. Section 3.1 constructs worker skill models when the labels Z are independent and Section 3.2 constructs worker skill models when the labels Z are dependent.

3.1. Independent Labels

Let x_i be the ith image, which contains an object with class label $y_i \in \{1, \dots, C\}$ (*e.g.*, species). Suppose a set of workers $\mathcal{W}_i$ independently specify their guess at the class of image i, such that for each $j \in \mathcal{W}_i$, z_{ij} is worker j's guess at y_i. In this situation, identifiers from Figure 1 would not get to observe preceding users' guesses. Let w_j be some set of parameters encoding worker j's skill at predicting classes. In this notation, if the class y_i is unknown, we can estimate the probability of each possible class given the set $Z_i = \{z_{ij}\}_{j \in \mathcal{W}_i}$ of worker guesses:

$$p(y_i|Z_i) = \frac{p(y_i) \prod_{j \in \mathcal{W}_i} p(z_{ij}|y_i, w_j)}{\sum_{y=1}^{C} p(y) \prod_{j \in \mathcal{W}_i} p(z_{ij}|y, w_j)} \tag{1}$$

where $p(y_i)$ is the prior class probability and $p(z_{ij}|y_i, w_j)$ is a model of imperfect human guesses. Sections 3.1.1-3.1.2 discuss possible models for $p(z_{ij}|y_i, w_j)$, which are also summarized in Table 1.

3.1.1 Flat Models

Flat Single Binomial: One simple way to model worker skills is with a single parameter that captures the worker's probability of providing a correct answer, regardless of the class label. We assume that the probability of worker being correct m_j follows a Bernoulli distribution, with other responses having probability proportional to class priors:

$$p(z_{ij}|y_i, w_j) = \begin{cases} m_j & \text{if } z_{ij} = y_i \\ (1 - m_j)p(z_{ij}) & \text{otherwise} \end{cases} \tag{2}$$

To prevent over fitting in low data situations, we place a beta prior Beta$(n_\beta p^c, n_\beta(1 - p^c))$ on m_j, where n_β is the strength of the prior. p^c represents the probability of any worker providing a correct label, and is estimated by pooling all worker annotations together. We also place a beta prior Beta$(n_\beta p, n_\beta(1 - p))$ on p^c, with p acting as our prior belief on worker performance. Estimating the worker skills is done by counting the number of times their response agrees with the predicted label, weighted by the prior strength:

$$m_j = \frac{n_\beta p^c + \sum_{i \in \mathcal{I}_j} 1[z_{ij} = \bar{y}_i, |\mathcal{W}_i| > 1] - 1}{n_\beta + \sum_{i \in \mathcal{I}_j} 1[\bar{y}_i, |\mathcal{W}_i| > 1] - 2} \tag{3}$$

where $1[\cdot]$ is the indicator function, $\mathcal{I}_j$ are the images labeled by worker j, and $\bar{y}_i$ is our current label prediction for image i. The pooled prior p^c is estimated similarly.

Flat Per Class Binomial: Rather than learning a single skill parameter m across all classes, we can learn a separate binomial model for each value of y, resulting in a skill vector M_j for each worker:

$$p(z_{ij}|y_i, w_j) = \begin{cases} M_j(y_i) & \text{if } z_{ij} = y_i \\ (1 - M_j(y_i))p(z_{ij}) & \text{otherwise} \end{cases} \tag{4}$$

Similar to the single binomial model, we employ a tiered prior system by adding a per class beta prior Beta$(n_\beta p^y, n_\beta(1-p^y))$ on $M_j(y)$. We place a generic beta prior Beta$(n_\beta p, n_\beta(1-p))$ on p^y to encode our prior belief that a worker is correct on any class. Estimating the worker skill parameters $M_j(y)$ and the pooled priors p^y for class y is done in the same way as the single binomial model.

Flat Per Class Multinomial: A more sophisticated model of $p(z_{ij}|y_i, w_j)$ could assume w_j encodes a $C \times C$ confusion matrix $\mathbf{M}_j$, where an entry $\mathbf{M}_j(m, n)$ denotes person j's probability of predicting class n when the true class is m. Here, $p(z_{ij}|y_i, w_j) = \mathbf{M}_j(y_i, z_{ij})$; the model is assuming $p(z_{ij}|y_i = c, w_j)$ is a multinomial distribution with parameters $\boldsymbol{\mu}_j^c = [\mathbf{M}_j(c,1), ..., \mathbf{M}_j(c, C)]$ for each value of c. We will place Dirichlet priors Dir$(n_\beta \boldsymbol{\alpha}^c)$ on $\boldsymbol{\mu}_j^c$, where n_β is the strength of the prior, and $\boldsymbol{\alpha}^c$ is estimated by pooling across all workers. We will also place a Dirichlet prior Dir$(n_\beta \boldsymbol{\alpha})$ on $\boldsymbol{\alpha}^c$, with $\boldsymbol{\alpha}$ acting as a global hyper-parameter that provides the likelihood of any worker labeling a class correctly. Because the Dirichlet distribution is the conjugate prior of the multinomial distribution, the computation of each entry k from $1 \ldots C$ in the skill vector $\boldsymbol{\mu}_j^c$ for a single worker j and each class c is done by counting agreements:

$$\mu_{j,k}^c = \frac{n_\beta \alpha_k^c + \sum_{i \in \mathcal{I}_j} 1[z_{ij} = k, \bar{y}_i = k, |\mathcal{W}_i| > 1] - 1}{n_\beta \alpha_0^c + \sum_{i \in \mathcal{I}_j} 1[\bar{y}_i = k, |\mathcal{W}_i| > 1] - C} \tag{5}$$

Where $\alpha_0^c = \sum_k \alpha_k^c$. The pooled worker parameters $\boldsymbol{\alpha}^c$ are estimated in a similar way.

3.1.2 Taxonomic Models

Multinomial models are useful because they model commonly confused classes, however they have far more parameters than the binomial models. These models quickly become intractable as the total number of classes C gets large. For example, if there are 10^4 classes, we would be attempting to estimate a matrix $\mathbf{M}_j$ with 10^8 entries for each worker j. This is statistically and computationally intractable. However, when the number of classes gets large there often exists a taxonomy used to organize them (*e.g.* the Linnaean taxonomy for biological classification). We can use this taxonomy to reduce the number of parameters in a multinomial model.

Taxonomic Per Class Multinomial: We will assume a taxonomy of classes that is L levels deep, and associate a confusion matrix with each node in the taxonomy (*e.g.*, if we know the genus of an observation from iNaturalist, assume each worker has a confusion matrix among species within that genus). For the taxonomic model, let y_i^l denote the node in the taxonomy at level l that class y_i belongs to, such that y_i^0 is the root node and y_i^L is the leaf node (*i.e.*, species label). Similarly, let z_{ij}^l denote the node in the taxonomy at level l that class z_{ij} belongs to. In this model, $p(z_{ij}^l|y_i^l, w_j, y_i^{l-1} = z_{ij}^{l-1}) = \mathbf{M}_j^{y_i^{l-1}}(y_i^l, z_{ij}^l)$, where $\mathbf{M}_j^{y_i^{l-1}}$ is a confusion matrix associated with node y_i^{l-1} in the taxonomy; the assumption is that for each value of y_i^l, z_{ij}^l is multinomial with a vector $\mathbf{M}_j^{y_i^{l-1}}(y_i^l, :)$ of parameters of size equal to the number of child nodes. The term $y_i^{l-1} = z_{ij}^{l-1}$ denotes the condition that the parent node classification is known. Suppose, however, that worker j is wrong about both the species and genus. We must also model $p(z_{ij}^l|y_i^l, w_j, y_i^{l-1} \neq z_{ij}^{l-1})$. In our model we assume that worker j predicts each class z_{ij}^l with some probability irrespective of the true class (assumes $p(z_{ij}^l|y_i^l, w_j, y_i^{l-1} \neq z_{ij}^{l-1}) = \mathbf{N}_j^{z_{ij}^{l-1}}(z_{ij}^l)$ is multinomial with a parameter for each possible child node). The taxonomic model results in the following values that can be plugged into Equation 1:

$$p(z_{ij}|y_i, w_j) = \prod_{l=1}^{L} p(z_{ij}^l|y_i^l, w_j), \tag{6}$$

$$p(z_{ij}^l|y_i^l, w_j) = \begin{cases} \mathbf{M}_j^{y_i^{l-1}}(y_i^l, z_{ij}^l) \text{ if } y_i^{l-1} = z_{ij}^{l-1} \\ \mathbf{N}_j^{z_{ij}^{l-1}}(z_{ij}^l) \text{ otherwise} \end{cases} \tag{7}$$

Note that in totality, for each node n in the taxonomy, we have associated a confusion matrix $\mathbf{M}_j^n$ with a row for each child of n, and a vector of probabilities $\mathbf{N}_j^n$ with an entry for each child. If the taxonomy is relatively balanced, this is far fewer parameters than the flat multinomial model (linear in the number of classes rather than quadratic). To make estimating worker parameters more robust, we will again make use of a tiered system of priors (*e.g.*, Dirichlet priors on all multinomial parameters) that are computed by pooling across all workers at each node. However, if this is still too many parameters, we can fall back to modeling the probability that a person is correct as a binomial distribution with a parameter per child node (*i.e.* the **taxonomic per class binomial** model), or even just one parameter for all children (*i.e.* the **taxonomic single binomial** model), assuming other class responses $z_{ij}^l \neq y_i^l$ have probability proportional to their priors. See Table 1 for an overview of all models.

3.1.3 Taxonomic Predictions

Thus far, we have assumed that a worker always predicts a class of the finest possible granularity (*i.e.*, species level). An alternate UI can allow a worker to predict an internal node in the taxonomy if unsure of the exact class, *i.e.* applying the "hedging your bets" [6] method to human classifiers. In Figure 1, this would be akin to one of the identifiers specifying the family Ardeidae, which includes both Snowy Egret and Great Egret. Let $\text{level}(z_{ij})$ be the level of this prediction. Note that z_{ij}^l is valid only for $l \leq \text{level}(z_j)$. The taxonomic model in Section 3.1.2 works after an update of Equation 6 to $p(z_{ij}|y_i, w_j) = \prod_{l=1}^{\text{level}(z_{ij})} p(z_{ij}^l|y_i^l, w_j)$. This works even if different workers provide different levels of taxonomic predictions.

3.2. Dependent Labels

In Section 3.1 we assumed each worker independently guesses the class of image i. We now turn to the situation described in Figure 1: a user submits an observation x_i and an initial identification z_{i,j_i^1}, where j_i^t denotes the tth worker that labeled image i. A notification of the observation is sent to users that have subscribed to the taxa z_{i,j_i^1} or to that particular geographic region (the rest of the community is not explicitly notified but can find the observation when browsing the site). Each subsequent identifier $j_i^t, t > 1$ can see the details of the observation x_i and all identifications made by previous users $H_i^{t-1} = \{z_{i,j_i^1}, z_{i,j_i^2}, ..., z_{i,j_i^{t-1}}\}$. Users can assess the experience of a previous identifier j by viewing all of their observations X_j and all of their identifications Z_j. Additionally, users are able to discuss the identifications through comments.

In this setting, we can adapt Equation 1 to

$$\begin{aligned} p(y_i|Z_i) &= p(y_i|H_i^{|\mathcal{W}_i|}) \\ &= \frac{p(y_i)\prod_{t=1}^{|\mathcal{W}_i|} p(z_{i,j_i^t}|y_i, H_i^{t-1}, w_{j_i^t})}{\sum_{y=1}^{C} p(y)\prod_{t=1}^{|\mathcal{W}_i|} p(z_{i,j_i^t}|y, H_i^{t-1}, w_{j_i^t})} \end{aligned} \tag{8}$$

There are many possible choices for modeling $p(z_{i,j_i^t}|y_i, H_i^{t-1}, w_{j_i^t})$. The simplest option assumes each worker ignores all prior responses; *i.e.*, $p(z_{i,j_i^t}|y_i, H_i^{t-1}, w_{j_i^t}) = p(z_{i,j_i^t}|y_i, w_{j_i^t})$. In practice, however, worker j_i^t's response will probably be biased toward agreeing with prior responses H_i^{t-1}, making a prediction combining both evidence from analyzing prior responses and from observing the image itself. The weight of this evidence should increase with the number of prior responses and could vary based on worker j_i^t's assessment of other worker's skill levels. In our model, we assume that worker j_i^t weights each possible response z_{i,j_i^t} (worker j_i^t's perception of the class of image i) with a term $p_{j_i^t}(H_i^{t-1}|z_{i,j_i^t})$ (worker j_i^t's perception of the probability of prior responses given that class). $p(z_{i,j_i^t}|y_i, H_i^{t-1}, w_{j_i^t})$ can then be expressed as:

$$\begin{aligned} p(z_{i,j_i^t}|y_i, H_i^{t-1}, w_{j_i^t}) &= \frac{p(z_{i,j_i^t}, H_i^{t-1}|y_i, w_{j_i^t})}{p(H_i^{t-1}|y_i, w_{j_i^t})} \\ &= \frac{p(z_{i,j_i^t}|y_i, w_{j_i^t}) p_{j_i^t}(H_i^{t-1}|z_{i,j_i^t}, w_{j_i^t})}{\sum_z p(z|y_i, w_{j_i^t}) p_{j_i^t}(H_i^{t-1}|z, w_{j_i^t})} \end{aligned} \tag{9}$$

where $p(z_{i,j_i^t}|y_i, w_{j_i^t})$ is modeled using a method described in Section 3.1. Worker j_i^t might choose to treat each prior response as independent sources of information $p_{j_i^t}(H_i^{t-1}|z_{i,j_i^t}, w_{j_i^t}) = \prod_{s=1}^{t-1} p_{j_i^t}(z_{i,j_i^s}|z_{i,j_i^t}, w_{j_i^s}^{j_i^t})$ where we have used the notation w_k^j to denote parameters for worker j's perception of worker k's skill. Alternatively, worker j may choose to account for the fact that earlier responses were also biased by prior responses using similar assumptions as we made in Equation 9, resulting in a recursive definition/computation of $p_{j_i^t}(H_i^{t-1}|z_{i,j_i^t}, w_{j_i^t}) =$

$$\begin{cases} \dfrac{p_{j_i^t}(z_{i,j_i^{t-1}}|z_{i,j_i^t}, w_{j_i^{t-1}}^{j_i^t}) p_{j_i^{t-1}}(H_i^{t-2}|z_{i,j_i^{t-1}}, w_{j_i^{t-2}}^{j_i^{t-1}})}{\sum_z p_{j_i^t}(z|z_{i,j_i^t}, w_{j_i^{t-1}}^{j_i^t}) p_{j_i^{t-1}}(H_i^{t-2}|z, w_{j_i^{t-2}}^{j_i^{t-1}})} & \text{if } t > 1 \\ p_{j_i^t}(z_{i,j_i^{t-1}}|z_{i,j_i^t}, w_{j_i^{t-1}}^{j_i^t}) & \text{if } t = 1 \end{cases} \tag{10}$$

The last choice to make is how to model probabilities of the form $p_j(z_k|z_j, w_k^j)$ (*i.e.* worker j's perception of worker k's responses)? One model that keeps the number of parameters low is a binomial distribution: worker j assumes other workers are correct with probability ρ_j; when they are incorrect, they respond proportionally to class priors:

$$p_j(z_k|z_j, w_k^j) = \begin{cases} \rho_j & \text{if } z_k = z_j \\ (1 - \rho_j)p(z_j) & \text{otherwise} \end{cases} \tag{11}$$

Here, ρ_j is a learned parameter expressing worker j's trust in the responses of other workers.

4. Taking Pixels into Account

Rather than relying on class priors $p(y_i)$ we can make use of a computer vision model with parameters θ that can predict the probability of each class occurring in each image $x_i \in X$. This results in an update to equation 1, changing $p(y_i)$ to $p(y_i|x_i, \theta)$. We use a computer vision model similar to the general purpose binary computer vision system trained by Branson *et al.* [2]. We extract "PreLogit" features $\phi(x_i)$ from an Inception-v3 [30] CNN for each image i, and use these features (fixed for all iterations) to train the weights θ of a linear SVM (using a one-vs-rest strategy), followed by probability calibration using Platt scaling [25]. We use stratified cross-validation to construct training and validation splits that contain at least one sample from each class. This results in probability estimates $p(y_i|x_i, \theta) = \sigma(\gamma\, \theta \cdot \phi(x_i))$, where γ is the probability calibration scalar from Platt scaling, and $\sigma(\cdot)$ is the sigmoid function. Fine-tuning a CNN on each iteration would lead to better performance [1, 24, 38], but is out of scope.

5. Experiments

We evaluate the proposed models on data collected from paid workers through Amazon Mechanical Turk (MTurk) and from non-paid citizen scientists who are members of the Cornell Lab of Ornithology (Lab of O) or iNaturalist (iNat). We follow a similar evaluation protocol to [2] and use Algorithm 1 from that work to run the experiments. For models that assume worker labels are independent, we simulate multiple trials by adding worker labels in random order.

Method	Label Error Rate (%)
[7], [4]	27.78
Majority Vote	24.07
Flat Multinomial,[5], [36],[13]	11.11
Flat Multinomial-CV, [32], [40]*	10.19

Table 2: Label error rates of different worker skill models on the binary Bluebird dataset [36] after receiving *all* 4,212 annotations. Our methods (**Flat Multinomial**, and **Flat Multinomial-CV**) are competitive with other methods. *[40] mistakenly reported 10.09.

For lesion studies, we simply turn off parts of the model by preventing those parts from updating. The tag *prob-worker* means that a global prior is computed across all workers and per worker skill model was used, the tag *online* means that online crowdsourcing was used (with risk threshold parameter $\tau_\epsilon = .02$), and the tag *cv* means that computer vision probabilities were used instead of class priors.

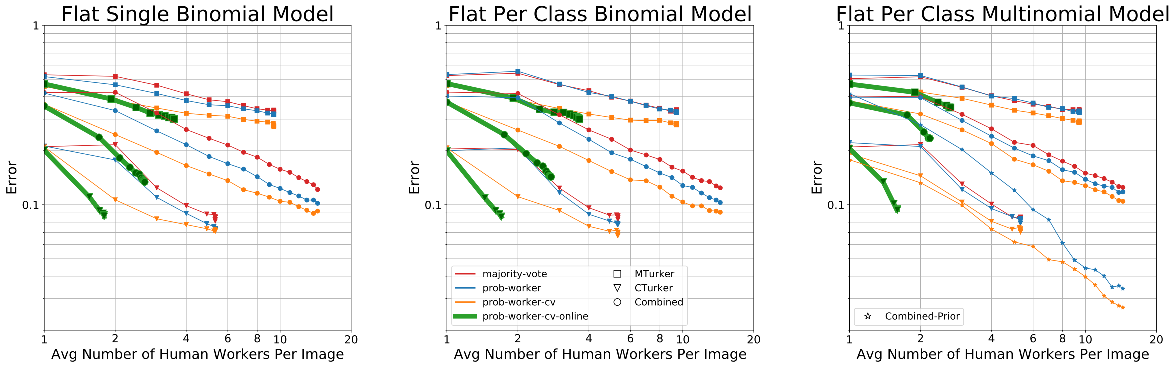
Bluebirds To gauge the effectiveness of our model against prior work, we run our models on the *binary* bluebird dataset from [36]. This dataset has a total of 108 images and 39 MTurkers labeled every image for a total of 4,212 annotations. Table 2 has the final label error rates of different worker skill models when *all* annotations are made available. Our offline, flat multinomial models are competitive with other offline methods.

NABirds This experiment was designed to test our models in a traditional dataset collection situation where labeling tasks are posted to a crowdsourcing website and responses are collected independently. We constructed a labeling interface that showed workers a sequence of 10 images and asked them to classify each image into one of 69 different bird species by using an auto complete box or by browsing a gallery of representative photos for each species. We used 998 images, all sampled from either shorebird or sparrow species, from the the NABirds dataset [34]. We collected responses from both MTurkers and citizen scientists from the Lab of O (CTurkers). Figure 3a shows the contribution of annotations from the workers. We had a total of 86 MTurkers provide 9,391 labels and a total of 202 CTurkers provide 5,300 labels. For these experiments we made the gallery of example images (3 to 5 images per species) available to the computer vision system during training. This ensured that we could construct at least 3 cross validation splits when calibrating the computer vision probabilities in the early stages of the algorithm.

All models were initialized with uniform class priors, a probability of 0.5 that an MTurker will label a class correctly, and a probability of 0.8 that a CTurker will label a class correctly. This means the global Dirichlet priors (used in the multinomial models) had a value of 0.8 at the true class index and 0.003 otherwise for the CTurkers. These are highly conservative priors. For each of our three flat models we conducted three experiments: using MTurk data only, using CTurk data only, and using both MTurk and CTurk data together ("Combined" in the plots). Figure 2 shows the results. First we note that when a computer vision system is utilized in an online fashion (prob-worker-cv-online) we see a significant decrease in the average number of labels per image to reach the same performance as majority vote using all of the data (*e.g.* a 5.4× decrease in the single binomial combined setting). In the offline setting (prob-worker-cv), the computer vision models decrease the final error compared to majority vote (*e.g.* 25% decrease in error in the single binomial combined setting). When considering our probabilistic model without computer vision (prob-worker) the single binomial model consistently achieved the lowest error, followed by the binomial per class model and then the multinomial model. This is not unexpected as we anticipated the larger capacity models to struggle with the sparseness of data (*i.e.* on average we had 0.75 labels per class per worker in the combined setting). However, the fact that they approach similar performance to the single binomial model highlights the usefulness of our tiered prior system and the ability to pool data across all of the workers. Our global prior initializations are purposefully on the conservative side, however in a real application setting, a user of this framework can initialize the priors using domain knowledge or a small amount of ground truth data. Figure 2c shows the dramatic effect of using more informative priors in the combined setting (prob-worker-cv and prob-worker in the Combined-Prior setting). These models were initialized with priors that were computed on a small held out set of worker annotations with ground truth labels and achieved the lowest error (0.03, for prob-worker-cv, a 79% decrease from majority vote) on the dataset.

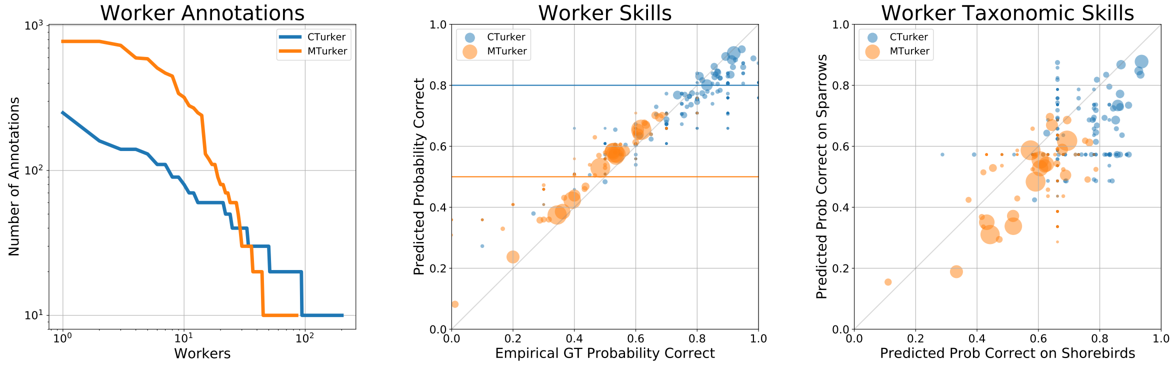
Figure 3b shows the predicted m_j values learned by the single binomial model plotted against the empirical ground truth in the combined setting. We can see that the model's predictions correlate well with the empirical estimates, with increasing precision as the number of annotations increases (size of the dots). To further investigate the worker skills we constructed a simple 2 level taxonomy and placed the shorebirds and sparrows in their own flat subtrees. By running our taxonomic binomial model we are able to learn a skill for each group separately, rendered in Figure 3c. We can see that both MTurkers and CTurkers have a higher probability of predicting shorebirds correctly than sparrows. In real applications we can use these skill estimates to direct images to proficient labelers.

iNaturalist This experiment was designed to test our models in a classification situation that mimics the real world scenario of websites like iNat, see Figure 1. We obtained a database export from iNat and cleaned the data using the following three steps: (1) We select observations and identifications from a subset of the taxonomy (*e.g.* species of



(a) (b) (c)

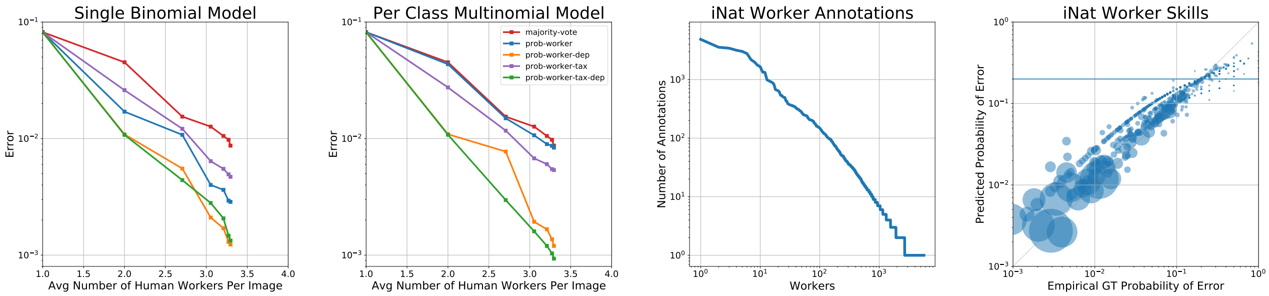
Figure 2: **Crowdsourcing Multiclass Labels with MTurkers and CTurkers:** These figures show results from our flat models on a dataset of 69 species of birds with labels from Amazon Mechanical Turk workers (MTukers) and citizen scientists (CTurkers). Each model was run on a dataset that consisted of: just MTurkers (squares), just CTurkers (triangles) or a combination of the two (circles). When our full framework is used (prob-worker-cv-online, green lines) we can achieve the same error as majority vote (red lines) with much fewer labels per image. When we use our framework in an offline setting (prob-worker-cv and prob-worker, orange and blue curves) we can achieve a lower error than majority vote with the same number of labels. When initialized with generic priors, the single binomial model achieves the lowest error, followed by the per class binomial and the multinomial model. However, if domain knowledge is used to initialize the global priors to more reasonable values, the multinomial model can achieve impressively low error (the star lines in **(c)**).



(a) (b) (c)

Figure 3: **MTurker and CTurker Worker Analysis:** Figure **(a)** shows the contribution of labels per worker from MTurkers and CTurkers. On average we have less than one label from each worker for each of the 69 classes, emphasizing the need to pool data across workers for use as priors. Figure **(b)** shows the predicted probability of a worker providing a correct label m_j plotted against the empirical ground truth probability for the single binomial prob-worker-cv model from 2a. The size of each dot is proportional to the number of annotations that worker contributed to the dataset. Solid lines mark the priors. We can see that the model's predictions correlate well with the empirical ground truths. Figure **(c)** shows the predicted worker skill for correctly labeling the species of a sparrow vs correctly labeling the species of a shorebird. These skill estimates came from a taxonomic binomial model with one subtree corresponding to sparrows and the other corresponding to shorebirds. In real applications we can use these skill estimates to direct images to proficient labelers.

birds). (2) For each observation, we keep only the first identification from each user (*i.e.* we do not allow users to change their minds). (3) To facilitate experiments, we keep all observations that have a ground truth label at the species level (*i.e.* leaf nodes of the taxonomy). For the experiments presented below, after performing the previous steps, we selected a subset of 30 species of birds and 1000 observations from each species to analyze. In this 30k image subset we have 5,643 workers that provided a total of 98,849 labels, Figure 4c shows the distribution of worker annotations. The taxonomy associated with these 30 species consisted of 44 nodes with a max depth of 3. For these experiments we did not utilize a computer vision system. Class priors were initialized to be uniform, skill priors were initialized assuming



(a) (b) (c) (d)

Figure 4: **iNaturalist Birds** Figures **(a)** and **(b)** show the errors achieved on a dataset of 30 bird species from iNaturalist for the single binomial and multinomial models respectively. Each model was evaluated in several configurations: "prob-worker" assumes a flat list of species. "prob-worker-tax" takes advantage of a taxonomy across the species, allowing workers to provide non-leaf node annotations and reducing the number of parameters in the multinomial model from 900 to 167. "prob-worker-dep" assumes a flat list of species, but models the dependence between the worker labels. "prob-worker-tax-dep" uses a taxonomy across the species and models the dependence between worker labels. All models did at least as well as majority vote, with dependence modeling providing a significant decrease in error. The lowest error was achieved by the multinomial prob-worker-tax-dep model that was capable of modeling species confusions and label dependencies, decreasing error by 90% compared to majority vote. Figure **(c)** shows the distribution of labels per worker, emphasizing a long tail of worker contributions. Figure **(d)** shows the predicted probability of error $(1 - m_j)$ for each worker plotted against the empirical ground truth probability of error for the single binomial prob-worker-dep model, with the radius of a dot proportional to the number of annotations contributed by that worker. The solid blue line is the global prior value. More active identifiers are less likely to make errors and our model skill estimates correlate well with the empirical ground truths.

that iNat users are 80% correct. Worker labels are added to the images sequentially by their time stamp, so only a single pass through the data is possible.

Figures 4a and 4b show the results for our single binomial and multinomial models respectively. For each model we used flat and taxonomic (-tax) versions, and we turned on (-dep) and off label dependence modeling, for a total of 4 variations of each model. We can see that all of our models are at least as good as majority vote. Adding dependence modeling to the flat models provides a significant decrease in error: a 59% decrease for the flat single binomial model, and a 85% decrease for the flat multinomial model. The taxonomic single binomial model (with 14 parameters per worker) did slightly worse than the flat single binomial model (with 1 parameter per worker). However, the taxonomic multinomial model (with 167 parameters per worker) decreased error by 36% compared to the flat multinomial model (with 900 parameters per worker). Finally, adding dependence modeling to the taxonomic models provided a further decrease in error, with the taxonomic multinomial model performing the best and decreasing error by 90% over majority vote, corresponding to 28 total errors. While a majority of those errors were true mistakes, an inspection of a few revealed errors in the ground truth labels of the iNat dataset. Figure 1 is actually an example of one of those mistakes. Further, the observation (https://tinyurl.com/ycu92cas) associated with the second "riskiest" image (using the computed Bayes risk of the predicted label $\mathcal{R}(\bar{y}_i)$) turned out to be another mistake, advocating the use of these models as a way of sorting the observations for expert review. Figure 4d shows the predicted probability of a worker labeling incorrectly $(1-m_j)$ for the flat single binomial model with dependence modeling from Figure 4a. We can see that the model's skill predictions correlate well with the empirical ground truth skills.

6. Conclusion

We introduced new multiclass annotation models that can be used in the online crowdsourcing framework of Branson *et al.* [2]. We explored several variants of a worker skill model using a variety of parameterizations and we showed how to harness a taxonomy to reduce the number of parameters when the number of classes is large. As an additional benefit, our taxonomic models are capable of processing worker labels from anywhere in the taxonomy rather than just leaf nodes. Finally, we presented techniques for modeling the dependence of worker labels in tasks where workers can see a prior history of identifications. Our models consistently outperform majority vote, either reaching a similar error with far fewer annotations or achieving a lower error with the same number of annotations. Future work involves modeling "schools of thought" among workers and using their skill estimates to explore human teaching.

Acknowledgments This work was supported by a Google Focused Research Award. We thank Oisin Mac Aodha for useful discussions.

References

[1] P. Agrawal, R. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *European conference on computer vision*, pages 329–344. Springer, 2014.

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

[3] G. Chen, S. Zhang, D. Lin, H. Huang, and P. A. Heng. Learning to aggregate ordinal labels by maximizing separating width. In *International Conference on Machine Learning*, pages 787–796, 2017.

[4] N. Dalvi, A. Dasgupta, R. Kumar, and V. Rastogi. Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd international conference on World Wide Web*, pages 285–294. ACM, 2013.

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

[6] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3450–3457. IEEE, 2012.

[7] A. Ghosh, S. Kale, and P. McAfee. Who moderates the moderators?: crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 167–176. ACM, 2011.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[9] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[10] R. Jin and Z. Ghahramani. Learning with multiple labels. In *Advances in neural information processing systems*, pages 897–904, 2002.

[11] E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 467–474. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[12] D. R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, pages 1953–1961, 2011.

[13] D. R. Karger, S. Oh, and D. Shah. Efficient crowdsourcing for multi-class labeling. *ACM SIGMETRICS Performance Evaluation Review*, 41(1):81–92, 2013.

[14] D. R. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

[15] A. Kovashka, O. Russakovsky, L. Fei-Fei, and K. Grauman. Crowdsourcing in Computer Vision. *ArXiv e-prints*, Nov. 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] H. Li, B. Yu, and D. Zhou. Error rate analysis of labeling by crowdsourcing. In *ICML Workshop: Machine Learning Meets Crowdsourcing. Atalanta, Georgia, USA*, 2013.

[18] G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 68–76. ACM, 2010.

[19] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[20] Q. Liu, J. Peng, and A. T. Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 692–700, 2012.

[21] C. Long, G. Hua, and A. Kapoor. Active visual recognition with expertise estimation in crowdsourcing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3000–3007, 2013.

[22] C. Mora, D. P. Tittensor, S. Adl, A. G. Simpson, and B. Worm. How many species are there on earth and in the ocean? *PLoS biology*, 9(8):e1001127, 2011.

[23] J. Ok, S. Oh, J. Shin, and Y. Yi. Optimality of belief propagation for crowdsourced classification. *arXiv preprint arXiv:1602.03619*, 2016.

[24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[25] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

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

[27] N. B. Shah and D. Zhou. Double or nothing: Multiplicative incentive mechanisms for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 1–9, 2015.

[28] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of venus images. 1995.

[29] B. L. Sullivan, J. L. Aycrigg, J. H. Barry, R. E. Bonney, N. Bruns, C. B. Cooper, T. Damoulas, A. A. Dhondt, T. Dietterich, A. Farnsworth, et al. The ebird enterprise: an integrated approach to development and application of citizen science. *Biological Conservation*, 169:31–40, 2014.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[31] W. Tang and M. Lease. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR 2011 workshop on crowdsourcing for information retrieval (CIR)*, pages 1–6, 2011.

[32] T. Tian and J. Zhu. Max-margin majority voting for learning from crowds. In *Advances in Neural Information Processing Systems*, pages 1621–1629, 2015.

[33] K. Ueda. iNaturalist Research-grade Observations via GBIF.org. *https://doi.org/10.15468/ab3s5x*, 2017.

[34] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.

[35] A. Vempaty, L. R. Varshney, and P. K. Varshney. Reliable crowdsourcing for multi-class labeling using coding theory. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):667–679, 2014.

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

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

[38] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[39] J. Zhang, V. S. Sheng, J. Wu, and X. Wu. Multi-class ground truth inference in crowdsourcing with clustering. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):1080–1085, 2016.

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

[41] D. Zhou, S. Basu, Y. Mao, and J. C. Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203, 2012.

[42] D. Zhou, Q. Liu, J. C. Platt, and C. Meek. Aggregating ordinal labels from crowds by minimax conditional entropy. In *ICML*, pages 262–270, 2014.