Online crowdsourcing: rating annotators and obtaining cost-effective labels

Peter Welinder Pietro Perona
California Institute of Technology
{welinder,perona}@caltech.edu

Abstract

Labeling large datasets has become faster, cheaper, and easier with the advent of crowdsourcing services like Amazon Mechanical Turk. How can one trust the labels obtained from such services? We propose a model of the labeling process which includes label uncertainty, as well a multi-dimensional measure of the annotators' ability. From the model we derive an online algorithm that estimates the most likely value of the labels and the annotator abilities. It finds and prioritizes experts when requesting labels, and actively excludes unreliable annotators. Based on labels already obtained, it dynamically chooses which images will be labeled next, and how many labels to request in order to achieve a desired level of confidence. Our algorithm is general and can handle binary, multi-valued, and continuous annotations (e.g. bounding boxes). Experiments on a dataset containing more than 50,000 labels show that our algorithm reduces the number of labels required, and thus the total cost of labeling, by a large factor while keeping error rates low on a variety of datasets.

1. Introduction

Crowdsourcing, the act of outsourcing work to a large crowd of workers, is rapidly changing the way datasets are created. Not long ago, labeling large datasets could take weeks, if not months. It was necessary to train annotators on custom-built interfaces, often in person, and to ensure they were motivated enough to do high quality work. Today, with services such as Amazon Mechanical Turk (MTurk), it is possible to assign annotation jobs to hundreds, even thousands, of computer-literate workers and get results back in a matter of hours. This opens the door to labeling huge datasets with millions of images, which in turn provides great possibilities for training computer vision algorithms.

The quality of the labels obtained from annotators varies. Some annotators provide random or bad quality labels in the hope that they will go unnoticed and still be paid, and yet others may have good intentions but completely misunderstand the task at hand. The standard solution to the problem of "noisy" labels is to assign the same labeling task to many different annotators, in the hope that at least a few of them will provide high quality labels or that a consensus emerges from a great number of labels. In either case, a large number of labels is necessary, and although a single label is cheap, the costs can accumulate quickly.

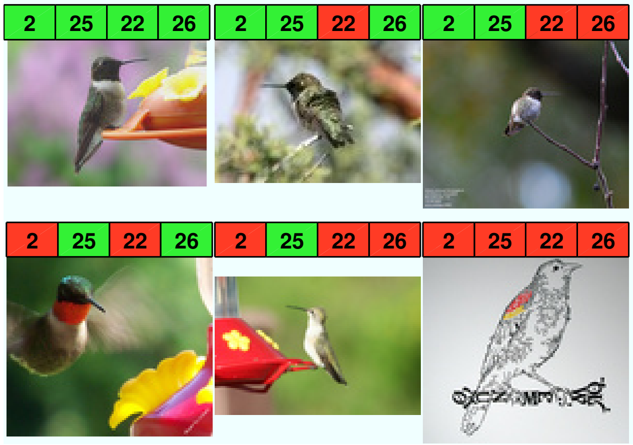

Figure 1. Examples of binary labels obtained from Amazon Mechanical Turk, (see Figure 2 for an example of continuous labels). The boxes show the labels provided by four workers (identified by the number in each box); green indicates that the worker selected the image, red means that he or she did not. The task for each annotator was to select only images that he or she thought contained a Black-chinned Hummingbird. Figure 5 shows the expertise and bias of the workers. Worker 25 has a high false positive rate, and 22 has a high false negative rate. Worker 26 provides inconsistent labels, and 2 is the annotator with the highest accuracy. Photos in the top row were classified to contain a Black-chinned Hummingbird by our algorithm, while the ones in the bottom row were not.

If one is aiming for a given label quality for the minimum time and money, it makes more sense to dynamically decide on the number of labelers needed. If an expert annotator provides a label, we can probably rely on it being of high quality, and we may not need more labels for that particular task. On the other hand, if an unreliable annotator provides a label, we should probably ask for more labels until we find an expert or until we have enough labels from non-experts to let the majority decide the label.

We present an online algorithm to estimate the reliability or expertise of the annotators, and to decide how many labels to request per image based on who has labeled it. The model is general enough to handle many different types of annotations, and we show results on binary, multi-valued, and continuous-valued annotations collected from MTurk.

The general annotator model is presented in Section 3 and the online version in Section 4. Adaptations of the model to discrete and continuous annotations are discussed in Section 5. The datasets are described in Section 6, the experiments in Section 7, and we conclude in Section 8.

2. Related Work

The quality of crowdsourced labels (also called annotations or tags) has been studied before in different domains. In computer vision, the quality of annotations provided by MTurk workers and by volunteers has been explored for a wide range of annotation types [8, 4]. In natural language processing, Snow et al. [7] gathered labels from MTurk and compared the quality to ground truth.

The most common method for obtaining ground truth annotations from crowdsourced labels is by applying a majority consensus heuristic. This has been taken one step further by looking at the consistency between labelers. For multi-valued annotations, Dawid and Skene [1] modeled the individual annotator accuracy by a confusion matrix. Sheng et al. [5] also modeled annotator quality, and showed how repeated and selective labeling increased the overall labeling quality on synthetic data. Smyth et al. [6] integrated the opinions of many experts to determine a gold standard, and Spain and Perona [9] developed a method for combining prioritized lists obtained from different annotators. Using annotator consistency to obtain ground truth has also been used in the context of paired games and CAPTCHAs [11, 12]. Whitehill et al. [14] consider the difficulty of the task and the ability of the annotators. Annotator models have also been used to train classifiers with noisy labels [3].

Vijayanarasimhan and Grauman [10] proposed a system which actively asks for image labels that are the most informative and cost effective. To our knowledge, the problem of online estimation of annotator reliabilities has not been studied before.

3. Modeling Annotators and Labels

We assume that each image i has an unknown "target value" which we denote by $\mathbf{z}_i$. This may be a continuous or discrete scalar or vector. The set of all N images, indexed by image number, is $\mathcal{I} = \{1, \ldots, N\}$, and the set of corresponding target values is abbreviated $\mathbf{z} = \{\mathbf{z}_i\}_{i=1}^N$. The reliability or expertise of annotator j is described by a vector of parameters, $\mathbf{a}_j$. For example, it can be scalar, $\mathbf{a}_j = a_j$, such as the probability that the annotator provides a correct label; specific annotator parameterizations are discussed in Section 5. There are M annotators in total, $\mathcal{A} = \{1, \ldots, M\}$, and the set of their parameter vectors is $\mathbf{a} = \{\mathbf{a}_j\}_{j=1}^M$. Each annotator j provides labels $\mathcal{L}^j = \{\mathbf{l}_{ij}\}_{i \in \mathcal{I}_j}$ for all or a subset of the images, $\mathcal{I}_j \subseteq \mathcal{I}$. Likewise, each image i has labels $\mathcal{L}_i = \{\mathbf{l}_{ij}\}_{j \in \mathcal{A}_i}$ provided by a subset of the annotators $\mathcal{A}_i \subseteq \mathcal{A}$. The set of all labels is denoted $\mathcal{L}$. For simplicity, we will assume that the labels $\mathbf{l}_{ij}$ belong to the same set as the underlying target values $\mathbf{z}_i$; this assumption could, in principle, be relaxed.

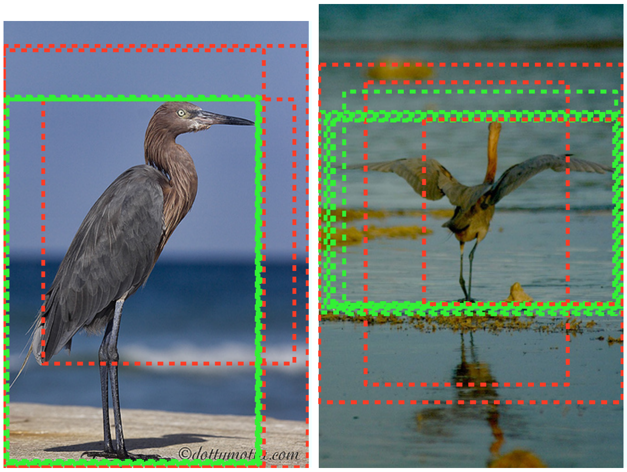

Figure 2. Examples of bounding boxes (10 per image) obtained from MTurk workers who were instructed to provide a snug fit. Per our model, the green boxes are correct and the red boxes incorrect. Most workers provide consistent labels, but two unreliable workers stand out: no. 53 and 58 (they provided two of the incorrect boxes in each image). As can be seen from Figure 6c, most of the labels provided by these two workers were of low quality.

Our causal model of the labeling process is shown schematically in Figure 3. The image target values and annotator parameters are assumed to be generated independently. To ensure that the estimation process degrades gracefully with little available label data, we take the Bayesian point of view with priors on $\mathbf{z}_i$ and $\mathbf{a}_j$ parameterized by $\boldsymbol{\zeta}$ and $\boldsymbol{\alpha}$ respectively. The priors encode our prior belief of how skilled the annotators are (e.g. that half will be experts and the rest unskilled), and what kind of target values we expect. The joint probability distribution can thus be factorized as

$$p(\mathcal{L}, \mathbf{z}, \mathbf{a}) = \prod_{i=1}^{N} p(\mathbf{z}_i \mid \boldsymbol{\zeta}) \prod_{j=1}^{M} p(\mathbf{a}_j \mid \boldsymbol{\alpha}) \prod_{\mathbf{l}_{ij} \in \mathcal{L}} p(\mathbf{l}_{ij} \mid \mathbf{z}_i, \mathbf{a}_j). \quad (1)$$

Inference: Given the observed variables, that is, the labels $\mathcal{L}$, we would like to infer the hidden variables, i.e. the target values $\mathbf{z}$, as well as the annotator parameters $\mathbf{a}$. This

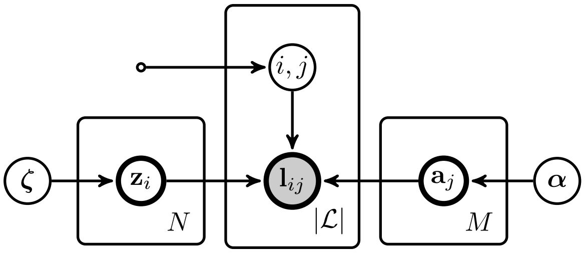

Figure 3. Plate representation of the general model. The i, j pair in the middle plate, indicating which images each annotator labels, is determined by some process that depends on the algorithm (see Sections 3–4).

can be done using a Bayesian treatment of the Expectation-Maximization (EM) algorithm [2].

E-step: Assuming that we have a current estimate $\hat{\mathbf{a}}$ of the annotator parameters, we compute the posterior on the target values:

$$\hat{p}(\mathbf{z}) = p(\mathbf{z} \mid \mathcal{L}, \hat{\mathbf{a}}) \propto p(\mathbf{z})\, p(\mathcal{L} \mid \mathbf{z}, \hat{\mathbf{a}}) = \prod_{i=1}^{N} \hat{p}(\mathbf{z}_i), \quad (2)$$

where

$$\hat{p}(\mathbf{z}_i) = p(\mathbf{z}_i \mid \boldsymbol{\zeta}) \prod_{j \in \mathcal{A}_i} p(\mathbf{l}_{ij} \mid \mathbf{z}_i, \hat{\mathbf{a}}_j). \quad (3)$$

M-step: To estimate the annotator parameters $\mathbf{a}$, we maximize the expectation of the logarithm of the posterior on $\mathbf{a}$ with respect to $\hat{p}(\mathbf{z}_i)$ from the E-step. We call the auxiliary function being maximized $Q(\mathbf{a}, \hat{\mathbf{a}})$. Thus the optimal $\mathbf{a}^\star$ is found from

$$\mathbf{a}^\star = \arg\max_{\mathbf{a}} Q(\mathbf{a}, \hat{\mathbf{a}}), \quad (4)$$

where $\hat{\mathbf{a}}$ is the estimate from the previous iteration, and

$$\begin{aligned} Q(\mathbf{a}, \hat{\mathbf{a}}) &= \mathbb{E}_{\mathbf{z}} \left[\log p(\mathcal{L} \mid \mathbf{z}, \mathbf{a}) + \log p(\mathbf{a} \mid \boldsymbol{\alpha}) \right] \quad (5) \\ &= \sum_{j=1}^{M} Q_j(\mathbf{a}_j, \hat{\mathbf{a}}_j), \quad (6) \end{aligned}$$

where $\mathbb{E}_{\mathbf{z}}[\cdot]$ is the expectation with respect to $\hat{p}(\mathbf{z})$ and $Q_j(\mathbf{a}_j, \hat{\mathbf{a}}_j)$ is defined as

$$Q_j(\mathbf{a}_j, \hat{\mathbf{a}}_j) = \log p(\mathbf{a}_j \mid \boldsymbol{\alpha}) + \sum_{i \in \mathcal{I}_j} \mathbb{E}_{\mathbf{z}_i} \left[\log p(\mathbf{l}_{ij} \mid \mathbf{z}_i, \mathbf{a}_j) \right]. \quad (7)$$

Hence, the optimization can be carried out separately for each annotator, and relies only on the labels that the annotator provided. It is clear from the form of (3) and (7) that any given annotator is not required to label every image.

```
Input: Set of images $\mathcal{U}$ to be labeled
 1: Initialize $\mathcal{I}, \mathcal{L}, \mathcal{E}, \mathcal{B} = \{\emptyset\}$
 2: while $|\mathcal{I}| < |\mathcal{U}|$ do
 3:     Add $n$ images $\{i : i \in (\mathcal{U} \setminus \mathcal{I})\}$ to $\mathcal{I}$
 4:     for $i \in \mathcal{I}$ do
 5:         Compute $\hat{p}(\mathbf{z}_i)$ from $\mathcal{L}_i$ and $\mathbf{a}$
 6:         while $\max_{\mathbf{z}_i} \hat{p}(\mathbf{z}_i) < \tau$ and $|\mathcal{L}_i| < m$ do
 7:             Ask expert annotators $j \in \mathcal{E}$ to provide a label $\mathbf{l}_{ij}$
 8:             if no label $\mathbf{l}_{ij}$ is provided within time $T$ then
 9:                 Obtain label $\mathbf{l}_{ij}$ from some annotator $j \in (\mathcal{A}' \setminus \mathcal{B})$
10:             $\mathcal{L}_i \leftarrow \{\mathcal{L}_i \cup \mathbf{l}_{ij}\}$, $\mathcal{A} \leftarrow \{\mathcal{A} \cup \{j\}\}$
11:             Recompute $\hat{p}(\mathbf{z}_i)$ from updated $\mathcal{L}_i$ and $\mathbf{a}$
12:     Set $\mathcal{E}, \mathcal{B} = \{\emptyset\}$
13:     for $j \in \mathcal{A}$ do
14:         Estimate $\mathbf{a}_j$ from $\hat{p}(\mathbf{z}_i)$ by $\max_{\mathbf{a}_j} Q_j(\mathbf{a}_j, \hat{\mathbf{a}}_j)$
15:         if var$(\mathbf{a}_j) < \theta_v$ then
16:             if $\mathbf{a}_j$ satisfies an expert criterion then
17:                 $\mathcal{E} \leftarrow \{\mathcal{E} \cup \{j\}\}$
18:             else
19:                 $\mathcal{B} \leftarrow \{\mathcal{B} \cup \{j\}\}$
20: Output $\hat{p}(\mathbf{z})$ and $\mathbf{a}$
```

Figure 4. Online algorithm for estimating annotator parameters and actively choosing which images to label. The label collection step is outlined on lines 3–11, and the annotator evaluation step on lines 12–19. See Section 4 for details.

4. Online Estimation

The factorized form of the general model in (1) allows for an online implementation of the EM-algorithm. Instead of asking for a fixed number of labels per image, the online algorithm actively asks for labels only for images where the target value is still uncertain. Furthermore, it finds and prioritizes expert annotators and blocks sloppy annotators online. The algorithm is outlined in Figure 4 and discussed in detail in the following paragraphs.

Initially, we are faced with a set of images $\mathcal{U}$ with unknown target values $\mathbf{z}$. The set $\mathcal{I} \subseteq \mathcal{U}$ denotes the set of images for which at least one label has been collected and $\mathcal{L}$ is the set of all labels provided so far. Initially $\mathcal{I}$ and the set of all labels $\mathcal{L}$ are empty. We assume that there is a large pool of annotators $\mathcal{A}'$, of different and unknown ability, available to provide labels. The set of annotators that have provided labels so far is denoted $\mathcal{A} \subseteq \mathcal{A}'$ and is initially empty. We keep two lists of annotators: the *expert*-list, $\mathcal{E} \subseteq \mathcal{A}$, is a set of annotators who we trust to provide good labels, and the *bot*-list, $\mathcal{B} \subseteq \mathcal{A}$, are annotators that we know provide low quality labels and would like to exclude from further labeling. We call the latter list "bot"-list because the labels could as well have been provided by a robot choosing labels at random. The algorithm proceeds by iterating two steps until all the images have been labeled: (1) the label collection step, and (2) the annotator evaluation step.

Label collection step: $\mathcal{I}$ is expanded with n new images from $\mathcal{U}$. Next, the algorithm asks annotators to label the images in $\mathcal{I}$. First annotators in $\mathcal{E}$ are asked. If no annotator from $\mathcal{E}$ is willing to provide a label within a fixed amount of time T, the label is instead collected from an annotator in $(\mathcal{A}' \setminus \mathcal{B})$. For each image $i \in \mathcal{I}$, new labels $\mathbf{l}_{ij}$ are requested until either the posterior on the target value $\mathbf{z}_i$ is above a confidence threshold τ,

$$\max_{\mathbf{z}_i} \hat{p}(\mathbf{z}_i) > \tau, \tag{8}$$

or the number of labels $|\mathcal{L}_i|$ has reached a maximum cutoff m. It is also possible to set different thresholds for different $\mathbf{z}_i$'s, in which case we can trade off the costs of different kinds of target value misclassifications. The algorithm proceeds to the annotator evaluation step.

Annotator evaluation step: Since posteriors on the image target values $\hat{p}(\mathbf{z}_i)$ are computed in the label collection step, the annotator parameters can be estimated in the same manner as in the M-step in the EM-algorithm, by maximizing $Q_j(\mathbf{a}_j, \hat{\mathbf{a}}_j)$ in (7). Annotator j is put in either $\mathcal{E}$ or $\mathcal{B}$ if a measure of the variance of $\mathbf{a}_j$ is below a threshold,

$$\text{var}(\mathbf{a}_j) < \theta_v, \tag{9}$$

where θ_v is the threshold on the variance. If the variance is above the threshold we do not have enough evidence to consider the annotator to be an expert or a bot (unreliable annotator). If the variance is below the threshold, we place the annotator in $\mathcal{E}$ if $\mathbf{a}_j$ satisfies some expert criterion based on the annotation type, otherwise the annotator will be placed in $\mathcal{B}$ and excluded labeling in the next iteration.

On MTurk the expert- and bot-lists can be implemented by using "qualifications". A qualification is simply a pair of two numbers, a unique qualification id number and a scalar qualification score, that can be applied to any worker. The qualifications can then be used to restrict (by inclusion or exclusion) which workers are allowed to work on a particular task.

5. Annotation Types

Binary annotations are often used for classification, such as "Does the image contain an object from the visual class X or not?". In this case, both the target value $\mathbf{z}_i$ and the label $\mathbf{l}_{ij}$ are binary (dichotomous) scalars that can take values $z_i, l_{ij} \in \{0, 1\}$. A natural parameterization of the annotators is in terms of true negative and true positive rates. That is, let $\mathbf{a}_j = (a_j^0, a_j^1)^{\mathrm{T}}$ be the vector of annotator parameters, where

$$\begin{aligned} p(l_{ij} = 1 \mid z_i = 1, \mathbf{a}_j) &= a_j^1, \\ p(l_{ij} = 0 \mid z_i = 0, \mathbf{a}_j) &= a_j^0. \end{aligned} \tag{10}$$

As a prior for $\mathbf{a}_j$ we chose a mixture of beta distributions,

$$p(a_j^0, a_j^1) = \sum_{k=1}^{K} \pi_k^a \,\text{Beta}(\alpha_{k,0}^0, \alpha_{k,0}^1)\, \text{Beta}(\alpha_{k,1}^0, \alpha_{k,1}^1). \tag{11}$$

Our prior belief of the number of different types of annotators is encoded in the number of components K. For example, we can assume $K = 2$ kinds of annotators: honest annotators of different grades (unreliable to experts) are modeled by Beta densities that are increasingly peaked towards one, and adversarial annotators who provide labels that are opposite of the target value are modeled by Beta distributions that are peaked towards zero (we have actually observed such annotators). The prior also acts as a regularizer in the EM-algorithm to ensure we do not classify an annotator as an expert until we have enough evidence.

The parameterization of true positive and true negative rates allows us to cast the model in a signal detection theoretic framework [15], which provides a more natural separation of annotator bias and accuracy. Assume a signal x_i is generated in the head of the annotator as a result of some neural processing when the annotator is looking at image i. If the signal x_i is above a threshold t_j, the annotator chooses $l_{ij} = 1$, otherwise picking $l_{ij} = 0$. If we assume that the signal x_i is a random variable generated from one of two distributions, $p(x_i \mid z_i = k) \sim \mathcal{N}(\mu_j^k, \sigma^2)$, we can compute the annotator's sensitivity index d_j', defined as [15],

$$d_j' = \frac{|\mu_j^1 - \mu_j^0|}{\sigma}. \tag{12}$$

Notice that d_j' is a quantity representing the annotator's ability to discriminate images belonging to the two classes, while t_j is a quantity representing the annotator's propensity towards label 1 (low t_j) or label 0 (high t_j). By varying t_j and recording the false positive and false negative rates, we get the receiver operating characteristic (ROC curve) of the annotator. When $t_j = 0$ then the annotator is unbiased and will produce equal false positive and negative error rates of 50%, 31%, 15% and 6% for $d_j' = \{0, 1, 2, 3\}$ respectively. It is possible go between the two parameterizations if we assume that σ is the same for all annotators. For example, by assuming $\sigma = 1$, $\mu_j^0 = -d_j'/2$ and $\mu_j^1 = d_j'/2$, we can convert between the two representations using,

$$\begin{bmatrix} 1 & \frac{1}{2} \\ 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} t_j \\ d_j' \end{bmatrix} = \begin{bmatrix} \Phi^{-1}(a_j^0) \\ \Phi^{-1}(1 - a_j^1) \end{bmatrix}, \tag{13}$$

where $\Phi^{-1}(\cdot)$ is the inverse of the standard normal cumulative probability density function.

For binary labels, the stopping criterion in (8) has a very simple form. Consider the logarithm of the ratio of the posteriors,

$$R_i = \log \frac{p(z_i = 1 \mid \mathcal{L}_i, \mathbf{a})}{p(z_i = 0 \mid \mathcal{L}_i, \mathbf{a})} = \log \frac{p(z_i = 1)}{p(z_i = 0)} + \sum_{l_{ij} \in \mathcal{L}_i} R_{ij}, \tag{14}$$

where $R_{ij} = \log \frac{p(l_{ij}|z_i=1,\mathbf{a}_j)}{p(l_{ij}|z_i=0,\mathbf{a}_j)}$. Thus, every label l_{ij} provided for image i by some annotator j adds another positive or negative term R_{ij} to the sum in (14). The magnitude

$|R_{ij}|$ increases with d'_j, so that the opinions of expert annotators are valued more than unreliable ones. The criterion in (8) is equivalent to a criterion on the magnitude on the log ratio,

$$|R_i| > \tau' \text{ where } \tau' = \log \frac{\tau}{1-\tau}. \tag{15}$$

Observe that τ' could be different for positive and negative R_i. One would wish to have different thresholds if one had different costs for false alarm and false reject errors. In this case, the stopping criterion is equivalent to Wald's stopping rule for accepting or rejecting the null hypothesis in the Sequential Probability Ratio Test (SPRT) [13].

To decide when we are confident in an estimate of $\mathbf{a}_j$, in the online algorithm, we estimate the variance var$(\mathbf{a}_j)$ by fitting a multivariate Gaussian to the peak of $p(\mathbf{a}_j \mid \mathcal{L}, \mathbf{z})$. As a criterion for expertise, i.e. whether to add annotator j to $\mathcal{E}$, we use $d'_j > 2$ corresponding to a 15% error rate.

Multi-valued annotations where $z_i, l_{ij} \in \{1, \ldots, D\}$, can be modeled in almost the same way as binary annotations. A general method is presented in [1] for a full confusion matrix. However, we used a simpler model where a single a_j parameterizes the ability of the annotator,

$$p(l_{ij} = z_i \mid a_j) = a_j, \tag{16}$$
$$p(l_{ij} \neq z_i \mid a_j) = \frac{1-a_j}{D-1}.$$

Thus, the annotator is assumed to provide the correct value with probability a_j and an incorrect value with probability $(1 - a_j)$. Using this parameterization, the methods described above can be applied to the multi-valued (polychotomous) case.

Continuous-valued annotations are also possible. To make this section concrete, and for simplicity of notation, we will use bounding boxes, see Figure 2. However, the techniques used here can be extended to other types of annotations, such as object locations, segmentations, ratings, etc.

The image labels and target values are the locations of the upper left (x_1, y_1) and lower right (x_2, y_2) corners of the bounding box, and thus $\mathbf{z}_i$ and $\mathbf{l}_{ij}$ are 4-dimensional vectors of continuous variables $(x_1, y_1, x_2, y_2)^{\mathrm{T}}$. The annotator behavior is assumed to be governed by a single parameter $a_j \in [0, 1]$, which is the probability that the annotator attempts to provide an honest label. The annotator provides a "random" bounding box with probability $(1 - a_j)$. An honest label is assumed to be normally distributed from the target value

$$p(\mathbf{l}_{ij} \mid \mathbf{z}_i) = \mathcal{N}(\mathbf{l}_{ij} \mid \mathbf{z}_i, \mathbf{\Sigma}), \tag{17}$$

where $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$ is assumed to be a diagonal. One can take the Bayesian approach and have a prior on σ, and let it vary for different images. However, for simplicity we choose to keep σ fixed at 4 pixels (in screen coordinates). If the annotator decides not to provide an honest label, the label is assumed to be drawn from a uniform distribution,

$$p(\mathbf{l}_{ij} \mid \mathbf{z}_i) = \lambda_i^{-2}, \tag{18}$$

where λ_i is the area of image i (other variants, such as a very broad Gaussian, are also possible). The posterior on the label, used in the E-step in (2), can thus be written as a mixture,

$$p(\mathbf{l}_{ij} \mid \mathbf{z}_i, a_j) = a_j \mathcal{N}(\mathbf{l}_{ij} \mid \mathbf{z}_i, \mathbf{\Sigma}) + (1 - a_j) \frac{1}{\lambda^2}. \tag{19}$$

The prior on $\mathbf{z}_i$ is modeled by a uniform distribution over the image area, $p(\mathbf{z}_i) = \lambda_i^{-2}$, implying that we expect bounding boxes anywhere in the image. Similarly to the binary case, the prior on a_j is modeled as a Beta mixture,

$$p(a_j) = \sum_{k=1}^{K} \pi_k^a \text{Beta}(\alpha_k^0, \alpha_k^1), \tag{20}$$

to account for at different groups of annotators of different skills. We used two components, one for experts (peaked at high a_j) and another for unreliable annotators (broader, and peaked at a lower a_j).

In the EM-algorithm we approximate the posterior on $\mathbf{z}_i$ by a delta function,

$$\hat{p}(\mathbf{z}_i) = p(\mathbf{z}_i \mid \mathcal{L}_i, \hat{a}_j) = \delta(\hat{\mathbf{z}}_i), \tag{21}$$

where $\hat{\mathbf{z}}_i$ is the best estimate of $\mathbf{z}_i$, to avoid slow sampling to compute the expectation in the E-step. This approach works well in practice since $\hat{p}(\mathbf{z}_i)$ is usually very peaked around a single value of $\mathbf{z}_i$.

6. Datasets

Object Presence: To test the general model applied to binary annotations, we asked workers on MTurk to select images if they thought the image contained a bird of a certain species, see Figure 1. The workers were shown a few example illustrations of birds of the species in different poses. We collected labels for two different bird species, Presence-1 (Black-chinned Hummingbird) and Presence-2 (Reddish Egret), summarized in Table 1.

Attributes: As an example of a multi-valued annotation, we asked workers to pick one out of D mutually exclusive choices for the shape of a bird shown in a photograph (Attributes-1, $D = 14$) and for the color pattern of its tail (Attributes-2, $D = 4$). We obtained 5 labels per image for a total of 6,033 images, see Table 1.

Bounding Boxes: The workers were asked to draw a tightly fitting bounding box around the bird in each image (details in Table 1). Although it is possible to extend the model to multiple boxes per image, we ensured that there was exactly one bird in each image to keep things simple. See Figure 2 for some examples.

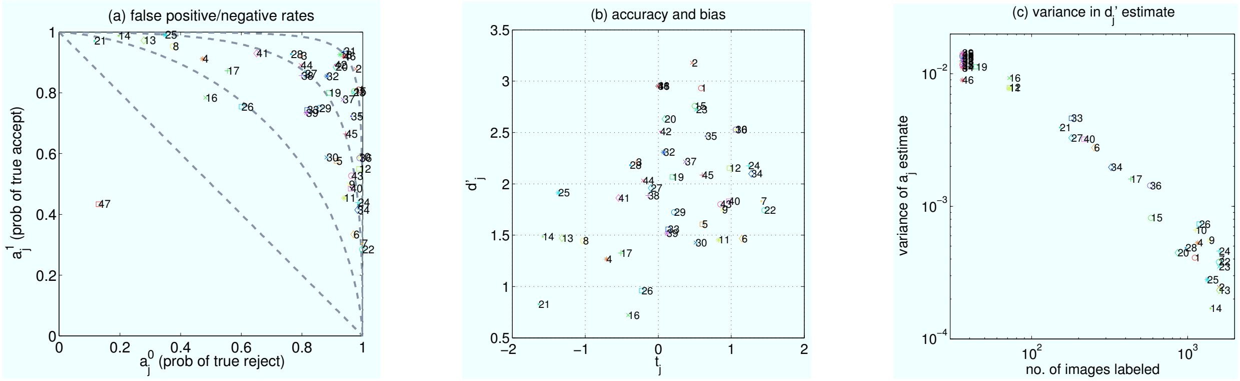

Figure 5. Estimates of expertise and bias in annotators providing binary labels. Annotators are plotted with different symbols and numbers to make them easier to locate across the plots. (a) Estimated true negative and positive rates (a_j^0, a_j^1). The dotted curves show the ROC curves for $d'_j = \{0, 1, 2, 3\}$. (b) Estimated t_j and d'_j from the data in (a). The accuracy of the annotator increases with d'_j and the bias is reflected in t_j. For example, if t_j is positive, the annotator has a high correct rejection rate at the cost of some false rejections, see Figure 1 for some specific examples. The outlier annotator, no. 47 in (a), with negative d'_j, indicating adversarial labels, was excluded from the plot. (c) The variance of the (a_j^0, a_j^1) decreases quickly with the number of images labeled. These diagrams show the estimates for the Presence-1 workers; Presence-2 gave very similar results.

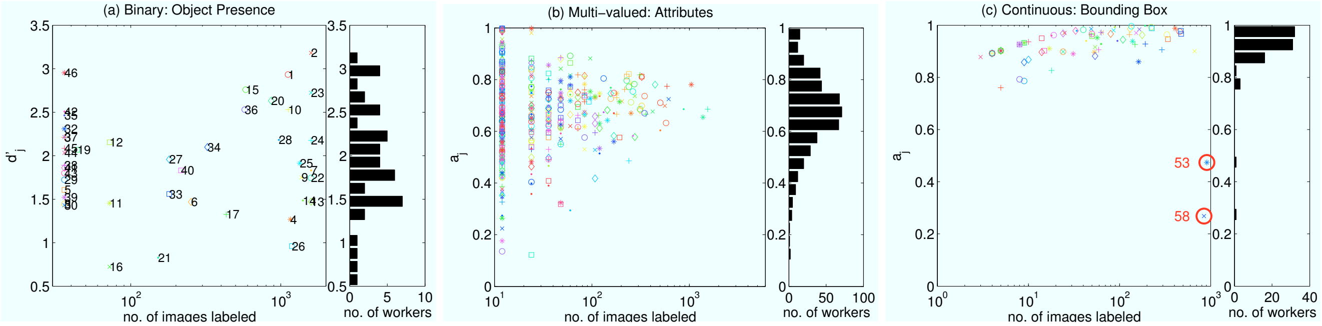

Figure 6. Quality index for each annotator vs. number of labeled images across the three annotation types. Annotators are plotted with different symbols and colors for easier visual separation. The bar chart to the right of each scatter plot is a histogram of the number of workers with a particular accuracy. (a) Object Presence: Shows results for Presence-1, Presence-2 is very similar. The minimum number of images a worker can label is 36, which explains the group of workers near the left edge. The adversarial annotator, no. 47, provided 36 labels and is not shown. (b) Attributes: results on Attributes-1. The corresponding plot for Attributes-2 is very similar. (c) Bounding box: Note that only two annotators, 53 and 58, labeled all 911 images. They also provided consistently worse labels than the other annotators. Figure 2 shows examples of the bounding boxes they provided.

7. Experiments and Discussion

To establish the skills of annotators on MTurk, we applied the general annotator model to the datasets described in Section 6 and Table 1. We first estimated $\mathbf{a}_j$ on the full datasets (which we call *batch*). We then estimated both the $\mathbf{a}_j$ and $\mathbf{z}_i$ using the online algorithm, as described in the last part of this section.

Annotator bias: The results of the batch algorithm applied to the Presence-1 dataset is shown in Figure 5. Different annotators fall on different ROC curves, with a bias towards either more false positives or false negatives. This is even more explicit in Figure 5b, where d'_j is a measure of expertise and t_j of the bias. What is clear from these figures is that most annotators, no matter their expertise, have some bias. Examples of bias for a few representative annotators and images are shown in Figure 1. Bias is something to keep in mind when designing annotation tasks, as the wording of a question presumably influences workers. In our experiments most the annotators seemed to prefer false negatives to false positives.

Annotator accuracy: Figure 6 shows how the accuracy of MTurk annotators varies with the number of images they label for different annotation types. For the Presence-1 dataset, the few annotators that labeled most of the available images had very different d'_j. For Attributes-1, on the other hand, the annotators that labeled most images have very similar a_j. In the case of the bounding box annota-

Dataset	Images	Assignments	Workers
Presence-1	1,514	15	47
Presence-2	2,401	15	54
Attributes-1	6,033	5	507
Attributes-2	6,033	5	460
Bounding Boxes	911	10	85

Table 1. Summary of the datasets collected from Amazon Mechanical Turk showing the number of images per dataset, the number of labels per image (assignments), and total number of workers that provided labels. Presence-1/2 are binary labels, and Attributes-1/2 are multi-valued labels.

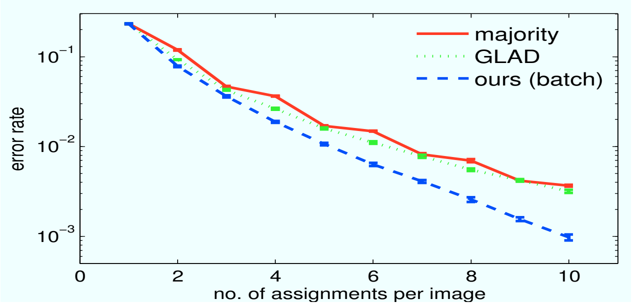

Figure 7. Comparison between the majority rule, GLAD [14], and our algorithm on synthetic data as the number of assignments per image is increased. The synthetic data is generated by the model in Section 5 from the worker parameters estimated in Figure 5a.

tions, most annotators provided good labels, except for no. 53 and 58. These two annotators were also the only ones to label all available images. In all three subplots of Figure 6, most workers provide only a few labels, and only some very active annotators label more than 100 images. Our findings in this figure are very similar to the results presented in Figure 6 of [7].

Importance of discrimination: The results in Figure 6 point out the importance of online estimation of $\mathbf{a}_j$ and the use of expert- and bot-lists for obtaining labels on MTurk. The expert-list is needed to reduce the number of labels per image, as we can be more sure of the quality of the labels received from experts. Furthermore, without the expert-list to prioritize which annotators to ask first, the image will likely be labeled by a new worker, and thus the estimate of $\mathbf{a}_j$ for that worker will be very uncertain. The bot-list is needed to discriminate against sloppy annotators that will otherwise annotate most of the dataset in hope to make easy money, as shown by the outliers (no. 53 and 58) in Figure 6c.

Performance of binary model: We compared the performance of the annotator model applied to binary data, described in Section 5, to two other models of binary data, as the number of available labels per image, m, varied. The first method was a simple majority decision rule and the second method was the GLAD-algorithm presented in [14]. Since we did not have access to the ground truth labels of the datasets, we generated synthetic data, where we knew the ground truth, as follows: (1) We used our model to estimate $\mathbf{a}_j$ for all 47 annotators in the Presence-1 dataset. (2) For each of 2000 target values (half with $z_i = 1$), we sampled labels from m randomly chosen workers, where the labels were generated according to the estimated $\mathbf{a}_j$ and Equation 10. As can be seen from Figure 7, our model achieves a consistently lower error rate on synthetic data.

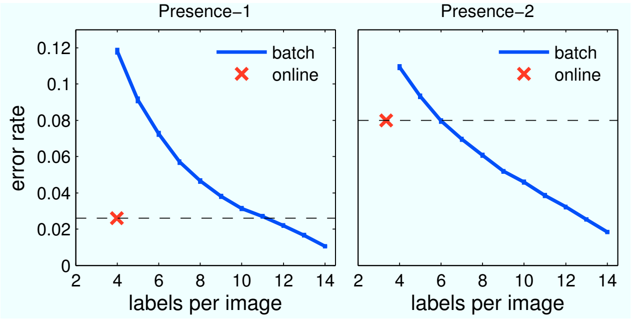

Figure 8. Error rates vs. the number of labels used per image on the Presence datasets for the online algorithm and the batch version. The ground truth was the estimates when running the batch algorithm with all 15 labels per image available (thus batch will have zero error at 15 labels per image).

Online algorithm: We simulated running the online algorithm on the Presence datasets obtained using MTurk and used the result from the batch algorithm as ground truth. When the algorithm requested labels for an image, it was given labels from the dataset (along with an identifier for the worker that provided it) randomly sampled without replacement. If it requested labels from the expert-list for a particular image, it only received such a label if a worker in the expert-list had provided a label for that image, otherwise it was randomly sampled from non bot-listed workers. A typical run of the algorithm on the Presence-1 dataset is shown in Figure 9. In the first few iterations, the algorithm is pessimistic about the quality of the annotators, and requests up to $m = 15$ labels per image. As the evidence accumulates, more workers are put in the expert- and bot-lists, and the number of labels requested by the algorithm decreases. Notice in the figure that towards the final iterations, the algorithm samples only 2–3 labels for some images.

To get an idea of the performance of the online algorithm, we compared it to running the batch version from Section 3 with limited number of labels per image. For the Presence-1 dataset, the error rate of the online algorithm is almost three times lower than the general algorithm when using the same number of labels per image, see Figure 8. For the Presence-2 dataset, twice as many labels per image are needed for the batch algorithm to achieve the same performance as the online version.

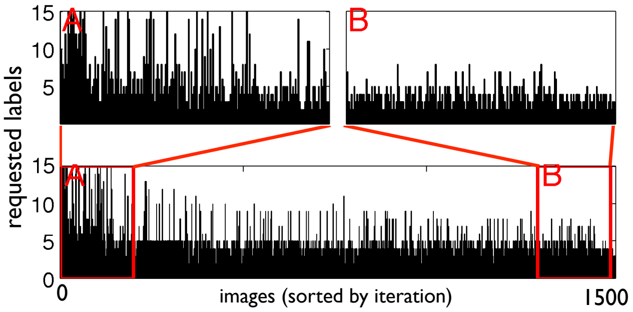

Figure 9. Progress of the online algorithm on a random permutation of the Presence-1 dataset. See Section 7 for details.

It is worth noting that most of the errors made by the online algorithm are on images where the intrinsic uncertainty of the ground truth label is high, i.e. $|R_i|$ as estimated by the full model using all 15 labels per image is large. Indeed, counting errors only for images where $|R_i| > 2$ (using log base 10), which includes 92% of the dataset, makes the error of the online algorithm drop to $0.75\% \pm 0.04\%$ on Presence-1. Thus, the performance clearly depends on the task at hand. If the task is easy, and most annotators agree, it will require few labels per image. If the task is difficult, such that even experts disagree, it will request many labels. The tradeoff between the number of labels requested and the error rate depends on the parameters used. Throughout our experiments, we used $m = 15$, $n = 20$, $\tau' = 2$, $\theta_v = 8 \times 10^{-3}$.

8. Conclusions

We have proposed an *online* algorithm to determine the "ground truth value" of some property in an image from multiple noisy annotations. As a by-product it produces an estimate of annotator expertise and reliability. It actively selects which images to label based on the uncertainty of their estimated ground truth values, and the desired level of confidence. We have shown how the algorithm can be applied to different types of annotations commonly used in computer vision: binary yes/no annotations, multi-valued attributes, and continuous-valued annotations (e.g. bounding boxes).

Our experiments on MTurk show that the quality of annotators varies widely in a continuum from highly skilled to almost random. We also find that equally skilled annotators differ in the relative cost they attribute to false alarm errors and to false reject errors. Our algorithm can estimate this quantity as well.

Our algorithm minimizes the labeling cost by assigning the labeling tasks preferentially to the best annotators. By combining just the right number of (possibly noisy) labels it defines an optimal 'virtual annotator' that integrates the real annotators without wasting resources. Thresholds in this virtual annotator may be designed optimally to trade off the cost of obtaining one more annotation with the cost of false alarms and the cost of false rejects. Future work includes dynamic adjustment of the price paid per annotation to reward high quality annotations and to influence the internal thresholds of the annotators.

Acknowledgments

We thank Catherine Wah, Florian Schroff, Steve Branson, and Serge Belongie for motivation, discussions and help with the data collection. We also thank Piotr Dollár, Merrielle Spain, Michael Maire, and Kristen Grauman for helpful discussions and feedback. This work was supported by ONR MURI Grant #N00014-06-1-0734 and ONR/Evolution Grant #N00173-09-C-4005.

References

[1] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *J. Roy. Statistical Society, Series C*, 28(1):20–28, 1979. 2, 5

[2] A. Dempster, N. Laird, D. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statistical Society, Series B*, 39(1):1–38, 1977. 3

[3] V. Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Valadez, L. Bogoni, and L. Moy. Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In *ICML*, 2009. 2

[4] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vis.*, 77(1–3):157–173, 2008. 2

[5] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, 2008. 2

[6] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of Venus images. *NIPS*, 1995. 2

[7] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, 2008. 2, 7

[8] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *First IEEE Workshop on Internet Vision at CVPR'08*, 2008. 2

[9] M. Spain and P. Perona. Some objects are more equal than others: measuring and predicting importance. In *ECCV*, 2008. 2

[10] S. Vijayanarasimhan and K. Grauman. What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *CVPR*, pages 2262–2269, 2009. 2

[11] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *SIGCHI conference on Human factors in computing systems*, pages 319–326, 2004. 2

[12] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008. 2

[13] A. Wald. Sequential tests of statistical hypotheses. *Ann. Math. Statist.*, 16(2):117–186, 1945. 5

[14] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2009. 2, 7

[15] T. D. Wickens. *Elementary signal detection theory*. Oxford University Press, United States, 2002. 4