A Full Probabilistic Model for Yes/No Type Crowdsourcing in Multi-Class Classification

Belen Saldias-Fuentes*† Pavlos Protopapas‡ Karim Pichara B.*‡

Abstract

Crowdsourcing has become widely used in supervised scenarios where training sets are scarce and difficult to obtain. Most crowdsourcing models in the literature assume labelers can provide answers to full questions. In classification contexts, full questions require a labeler to discern among all possible classes. Unfortunately, discernment is not always easy in realistic scenarios. Labelers may not be experts in differentiating all classes. In this work, we provide a full probabilistic model for a shorter type of queries. Our shorter queries only require "yes" or "no" responses. Our model estimates a joint posterior distribution of matrices related to labelers' confusions and the posterior probability of the class of every object. We developed an approximate inference approach, using Monte Carlo Sampling and Black Box Variational Inference, which provides the derivation of the necessary gradients. We built two realistic crowdsourcing scenarios to test our model. The first scenario queries for irregular astronomical time-series. The second scenario relies on the image classification of animals. We achieved results that are comparable with those of full query crowdsourcing. Furthermore, we show that modeling labelers' failures plays an important role in estimating true classes. Finally, we provide the community with two real datasets obtained from our crowdsourcing experiments. All our code is publicly available[1].

1 Introduction.

Labeled data is the very first requirement for training classifiers. Moreover, the availability of data has stimulated great breakthroughs in AI. For example, convolutional neural networks (CNNs) were first proposed by [16], but only when ImageNet [5] achieved a corpus of 1.5 million labeled images could Google's GoogLeNet [15] perform object classification almost as well as humans by using CNNs. This encouraged us to create new mechanisms for producing labels. Nevertheless, labeling means getting ground truths, which are often difficult, expensive, or impossible to obtain.

To increase the amount of labeled data, we can use crowdsourcing [4, 23, 27, 28] to gather a large amount of labels. A major challenge is to combine unreliable crowd information: this is not entirely accurate, but cheaper [32]. A typical case is to take the majority of votes for each object. For this to work, we must assume everyone has equal knowledge about the topic, which is in many cases a wrong assumption. In addition, we can use active learning (AL) [34, 31], a semi-supervised scenario in which a learning model iteratively selects the best instances (for example, those that most confuse the model) to be tagged by an expert. We can also mix these strategies [34, 17, 32] to select candidates by considering labelers' expertise. Nevertheless, here we propose a model to make the labeling task even easier.

Full **ABCD** query type:

What class of animal is this?	***Bird*** or ***Reptile*** or ***Mammal*** or ***Amphibian***

Proposed **YN** query type:

Is this animal a Bird?	***Yes*** or ***No***
Is this animal a Reptile?	***Yes*** or ***No***
Is this animal a Mammal?	***Yes*** or ***No***
Is this animal an Amphibian?	***Yes*** or ***No***

(a) Query types

*Questions for **Labeler 1**:*

Is this animal a Mammal?	***Yes*** or ***No***

*Questions for **Labeler 2**:*

Is this animal a Bird?	***Yes*** or ***No***
Is this animal a Mammal?	***Yes*** or ***No***
Is this animal a Reptile?	***Yes*** or ***No***

*Questions for **Labeler 3**:*

Is this animal a Bird?	***Yes*** or ***No***
Is this animal a Mammal?	***Yes*** or ***No***
Is this animal an Amphibian?	***Yes*** or ***No***
Is this animal a Reptile?	***Yes*** or ***No***

*Questions for **Labeler 4**:*

Is this animal a Bird?	***Yes*** or ***No***
Is this animal a Reptile?	***Yes*** or ***No***

(b) Questions for a crowd

Figure 1: Different query scenarios. Figure 1(a) shows the spotted salamander, an amphibian. Figure 1(b) shows a possible scenario with four labelers, four classes, and "yes" or "no" questions for the animal in 1(a).

Instead of selecting the best instances as candidates for training the model, we propose a novel approach to query type (see figure 1). Typically in a four-class scenario, a labeler is asked the class of an object with possible responses "A" or "B" or "C" or "D". We refer to that type of full question as an ABCD question. Our model generates low-cost queries in which each response gives partial information. This method iteratively selects, per labeler, a random object along with a class' label, then asks if that object belongs to

*Computer Science Department, Pontificia Universidad Católica de Chile, Santiago, Chile.

†MIT Media Lab, Massachusetts Institute of Technology, Cambridge, MA, USA (present affiliation, belen@mit.edu).

‡Institute for Applied Computational Science, Harvard University, Cambridge, MA, USA.

[1] `https://github.com/bcsaldias/yes-no-crowdsourcing`

that class: "yes" or "no" (proposed YN question).

The proposed method has many advantages over traditional approaches. First, the YN model focuses on the importance of learning an estimation of how labelers fail. Our strategy probabilistically learns initial parameters from the data for the labeling stage. Second, the labelers do not need to know all the classes. Third, it captures partial information with fewer errors because the labelers do not need to know the ground truth to accurately respond to some YN questions. Finally, the method is independent of the kind of data, given that we only need to include labelers' votes, without worrying about representation of the objects to be classified.

This work makes the following main contributions:

1. *Crowdsourcing query type*: We propose a new crowdsourcing framework to obtain labeled data focused on the query type. This method costs less than other models because it reconstructs ground truth labels by only using partial information. We show that the aggregation of partial information allows the YN model to ask fewer questions than others, while achieving similar accuracy.

2. *New data released*: We developed two real-world experiments with humans and published the data.

The rest of this paper is organized as follows: Section 2 presents some related work. In section 3 we explain the proposed model, and in section 4 we show how we solved it. Then, section 5 describes our implementations of the model. Section 6 describes the datasets for comparison. Then, section 7 shows experiments and analysis. Finally, in section 8 we discuss and conclude with the main results of our work.

2 Related Work

2.1 Creating Training Sets To acquire labels, we can manually label as many objects as possible. Furthermore, others have used crowdsourcing or/and active learning [34, 17, 32]. From another point of view, [22] proposes using *data programming*, in which labelers give functions that return the asked labels. Another option to create labels is *co-training* [1], in which data is labeled from two independent views. Closer to our approach is *boosting* [25], which combines several "weak" classifiers to create a "strong" one. We considered the weaknesses by modeling the labelers' (many views) errors to infer the true labels probabilistically.

2.2 Crowdourcing Scenarios Several efforts have been made on estimating labelers' expertise [33, 31, 17, 27] and maximizing labelers' accuracy by giving them the right incentives [26]. Some researchers have proposed new query types on active learning scenarios [21, 12]. Additionally, there are strategies to optimize the trade-off between redundancy and reliability in multi-class scenarios [13]. The closest research to the YN query type [19] involved assuming that each instance could belong to more than one class. However, these works did not involve a crowdsourcing context to improve the scenario. They mostly maintained a perfect oracle assumption.

Until now, no research has been presented to integrate query type, partial information asked to labelers, and the power of crowd. We propose a mechanism that outperforms other methods and handles many difficulties, as we outlined in section 1 and through this paper.

2.3 Variational Inference Approaches Several inference schemes have been used to solve the YN model. Following a probabilistic perspective, EM or MAP algorithms make the YN model very likely to converge to a local optimum [23, 32]. This can be handled using the Gibbs sampler [9, 17]. Previous research on labeling has always involved methods for full questions.

We used the No-U-Turn Hamiltonian sampler (NUTS) [11] to converge more quickly than the random walk that MCMC [10, 7] uses. Additionally, we tested Black Box variational inference (BBVI) [20] because it tends to be faster than NUTS [27]. BBVI is inexpensive and easy to implement because it only requires estimating the ELBO gradient.

3 The Model

Consider a dataset with $\mathcal{N}$ objects; each object $\mathcal{X}_i$ has only one true class z_i, among $\mathcal{K}$ possible classes, where $i \in \{1, ..., \mathcal{N}\}$ and $\mathbf{Z} = \{z_1, ..., z_i, ..., z_\mathcal{N}\}$. Each labeler $\mathcal{L}_j$ is then presented with a series of binary "yes" or "no" (YN) questions, where $j \in \{1, ..., \mathcal{J}\}$.

Formally, we define a YN question k_i^j as the question asked to labeler $\mathcal{L}_j$ about whether $\mathcal{X}_i$ belongs ("yes" or "no") to the class M_k, $k \in \{1, ..., \mathcal{K}\}$. We define $\mathcal{K}_i^j$ as the set of k_i^j queries asked to labeler $\mathcal{L}_j$ for the object $\mathcal{X}_i$. Let r_{ik}^j be the response (or vote) assigned by $\mathcal{L}_j$ to the question k_i^j, and $\mathbf{R}$ the set of all responses r_{ik}^j. Note that a labeler is not asked twice for the same class for the same object.

We propose a probabilistic graphical model [14, 29] (shown in figure 4) to infer the true labels $\mathbf{Z}$. The `Labeling` area represents the joint distribution of $\mathbf{Z}$ and the other variables involved in their prediction.

3.1 Responses For object $\mathcal{X}_i$, labeler $\mathcal{L}_j$, and question k_i^j, it is convenient to encode the response as a two dimensional vector: r_{ik}^j, where $[0, 1] \leftarrow$ [YES, NO]. Fig-

ure 2 shows an example of votes for object $\mathcal{X}_i$ given by labeler $\mathcal{L}_j$. Note that $r_{ik}^j = [0, 0]$ means that question k_i^j was not asked.

Question for	*Yes*	*No*
Class $\mathcal{M}_1$	1	0
Class $\mathcal{M}_2$	0	0
$\vdots$	$\vdots$	$\vdots$
Class $\mathcal{M}_k$	0	1
$\vdots$	$\vdots$	$\vdots$
Class $\mathcal{M}_\mathcal{K}$	0	1

Figure 2: Responses/votes r_i^j.

3.2 Credibility Matrices Common approaches involve the use of the confusion matrix of each labeler to represent their errors, due to the nature of the full question. We represented the YN error per labeler as a *credibility matrix*. We needed to find the probability per labeler of giving the right answer when the class asked is $M_{k'}$, and the true class is M_k. Figure 3 shows the credibility matrix of a specific labeler, where $\theta_{kk'}^j$ is the probability of labeler $\mathcal{L}_j$ saying "yes" to question k'^j_i when $z_i = k$. We assumed that the labelers were not random voters so that we could find patterns in their behaviors.

Our main goal was to find the most likely class for each object, given the votes and *credibility matrices* Θ. A side goal was to estimate Θ. In particular, we considered conjugate priors. Given that each "yes" or "no" $r_{kk'}^j$ response can be modeled as a Bernoulli distribution, the prior for $\theta_{kk'}^j$ distributes $\text{Beta}(\hat{\alpha}_{kk'}^j, \hat{\beta}_{kk'}^j)$, where $\hat{\alpha}_{kk'}^j$ and $\hat{\beta}_{kk'}^j$ are the estimated prior initial parameters from the first stage. Finally, the likelihood is:

$$r_{kk'}^j \sim \text{Bernoulli}(\theta_{kk'}^j)$$

Modeling the prior of $\theta_{kk'}^j$ as a Beta distribution that lives in a 0 to 1 space allowed us to model the probability of a response. It is also a conjugate distribution for the Bernoulli likelihood and can model any expertise due to its flexibility.

3.3 Joint Distribution Each YN vote r_{ik}^j depends on the real, but unknown, label z_i. Furthermore, the vote also depends on the credibility $\theta_{z_i k}^j$ of labeler $\mathcal{L}_j$. The conditioning to z_i allows the labeler to be more accurate in subsets of classes. The dependency on Θ^j allowed us to model the labeler's biases and errors for all classes. These dependencies are represented by the conditional distribution $\text{P}(r_{ik}^j | z_i, \theta_{z_i k}^j)$ [17].

	Question for $\mathcal{M}_{k'}$					
True Class $\mathcal{M}_k$	$\theta_{1,1}$	$\theta_{1,2}$	$\dots$	$\theta_{1,k'}$	$\dots$	$\theta_{1,\mathcal{K}}$
	$\theta_{2,1}$	$\theta_{2,2}$	$\dots$	$\theta_{2,k'}$	$\dots$	$\theta_{2,\mathcal{K}}$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
	$\theta_{k,1}$	$\theta_{k,2}$	$\dots$	$\theta_{k,k'}$	$\dots$	$\theta_{k,\mathcal{K}}$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
	$\theta_{\mathcal{K},1}$	$\theta_{\mathcal{K},2}$	$\dots$	$\theta_{\mathcal{K},k'}$	$\dots$	$\theta_{\mathcal{K},\mathcal{K}}$

Figure 3: Credibility matrix. Note that the rows are not required to sum 1.

From prior information, we could estimate the initial class proportions ρ and define a global Dirichlet variable π in charge of this unknown distribution of vector $\mathbf{Z}$. Finally, this gave:

$$\pi \sim \text{Dirichlet}(\rho)$$

$$z_i \sim \text{Categorical}(\pi)$$

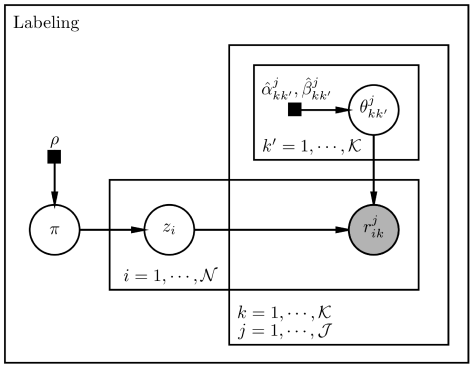

Figure 4: Proposed PGM. In this plate notation, random variables are clear circles; observed variables are shaded in gray. Estimated prior hyperparameters are represented by squares.

Likelihood We started from a single labeler, one object, and one question. For labeler $\mathcal{L}_j$ and question k_j^j, the likelihood is found in (3.1), where we encoded the response as a two-dimensional vector: r_{ik}^j, where $[0, 1] \leftarrow$ [YES, NO]. For all responses $\mathbf{R}$, all labelers $\mathcal{L}$, and all data $\mathcal{N}$, the likelihood is found in (3.2).

$$\text{P}(r_{ik}^j | \theta_{z_i k}^j, z_i) = \overbrace{(\theta_{z_i k}^j)^{r_{ik}^j[0]}}^{\substack{YES \\ r_{ik}^j[0]=1, r_{ik}^j[1]=0}} \times \overbrace{(1 - \theta_{z_i k}^j)^{r_{ik}^j[1]}}^{\substack{NO \\ r_{ik}^j[0]=0, r_{ik}^j[1]=1}} \tag{3.1}$$

$$\text{P}(\mathbf{R} | \Theta, \mathbf{Z}) \propto \prod_{i=1}^{\mathcal{N}} \prod_{j=1}^{\mathcal{J}} \prod_{k \in \mathcal{K}_i^j} \left\{ (\theta_{z_i k}^j)^{r_{ik}^j[0]} (1 - \theta_{z_i k}^j)^{r_{ik}^j[1]} \right\} \tag{3.2}$$

4 Inference Schema

We separated the inference into two intuitive stages: first, to estimate the labelers' reliability by asking them for known objects $\hat{\mathcal{N}}$ (Training Set), and second to ask them for unknown objects labels. We could unify these stages in a single inference model with an identical result. In the scenario where $\hat{\mathbf{Z}}$ are observed values, the model estimates beforehand $\hat{\Theta}$ and converges faster (see section 7). The likelihood for all responses $\hat{\mathbf{R}}$, all labelers $\mathcal{L}$, and all data $\hat{\mathcal{N}}$ is found in (4.3).

$$\text{(4.3)}\quad \mathrm{P}(\hat{\mathbf{R}}, \hat{\mathbf{Z}} \,|\hat{\Theta}) \propto \prod_{i=1}^{\hat{\mathcal{N}}} \prod_{j=1}^{\mathcal{J}} \prod_{k \in \hat{\mathcal{K}}_i^j} \left\{ (\hat{\theta}^j_{\hat{z}_i k})^{\hat{r}^j_{ik}[0]} \, (1 - \hat{\theta}^j_{\hat{z}_i k})^{\hat{r}^j_{ik}[1]} \right\}$$

The prior distribution of each $\hat{\theta}$ was chosen to be uninformative, but flexible enough to represent labelers with both high and low expertise. We selected $\hat{\theta}^j_{kk'} \sim \mathrm{Beta}(\alpha, \beta)$ with an expected value equivalent to 0.5 (see section 7). As stated before, this inference scheme works in two stages (that can also be done analytically):

1. `Credibility` stage: estimating $\hat{\Theta}$. Because we assumed the labelers would behave similarly in the `Labeling` stage, as they do here, we obtained the $\hat{\alpha}^j_{kk'}$ and $\hat{\beta}^j_{kk'}$ parameters from each $\hat{\theta}^j_{kk'}$.
2. `Labeling` stage: predicting $\mathbf{Z}$ and Θ via posterior inference.

5 Implementation

Due to the convergence time of NUTS, we also used BBVI [20], both in Python3.5. Each one works as follows: First, it estimates the latent variables $\hat{\Theta}$. Second, it estimates Θ, $\mathbf{Z}$, and π. All the experiments presented in section 7 used NUTS [24], except when indicated otherwise. BBVI approximately tries to find a probability distribution that is closest (in KL divergence) to the true posterior distribution. The supplementary material provides the derivation of the needed gradients to solve the model, which can be easily extended to any model with similar variable types (based on [2]).

6 Data

We used simulated and real-world datasets. First, we simulated data to understand the YN model's behavior. Then, we trained classifiers with real-world data to produce responses and evaluate the YN model performance. Finally, we tested the model in two human scenarios. These three sources of labels are described in the following subsections.

6.1 Synthetic Votes for Synthetic Data. To simulate labelers and their votes, we proceeded as follows: First, we created labels ($\hat{\mathbf{Z}}$ and $\mathbf{Z}$). Then, for each labeler, we sampled a credibility matrix. Each row was simulated using a $\mathrm{Beta}(0.5, 0.5)$ distribution. Labelers have high expertise in at most half of the classes; expertises were sampled from a $\mathrm{Beta}(20, 1)$ distribution (because its expected value is close to 1). Finally, we simulated the votes using the labelers and true labels. When the labeler $\mathcal{L}_j$ was presented with object $\mathcal{X}_i$ of class z_i and is asked k_i^j, we consulted its credibility matrix to obtain the response for k_i^j. We took r^j_{ik} by flipping a coin with the probability given by $\theta^j_{z_i k}$.

6.2 Synthetic Votes for Real-World Data. We used a subset of MACHO data [3] (250 objects). We trained six different classifiers as labelers, each with a different training set but equally sized (2 Random Forest classifiers, 2 Logistic Regressions, and 2 Support Vector Machines). We proceeded as follows: First, we split the data into three different sets; one to train classifiers, another to infer $\hat{\Theta}$, and the last to test the model. Each labeler was composed of a pool of $\mathcal{K}$ one-vs-all classifiers. When a labeler was asked for k_i^j, we consulted its one-vs-all binary classifier for the class $\mathcal{M}_k$ to get the probability of the object belonging to the class M_k. Then, we flipped a coin with that probability to obtain $\hat{\mathbf{R}}$ and $\mathbf{R}$.

MACHO data: Irregularly-sampled time series. Several works aim to classify astronomical irregular time series [18]. Table 1 shows the data distribution that we used.

6.3 Real Votes for Real-World Data. Two websites were set up to acquire data from human crowds. Each of them presented a contest to people related to a specific dataset domain (see table 1):

1. Astronomical irregular time series: We aim to classify irregular time series of the Catalina Surveys [6]. The labelers, 8 in total, were astronomers and engineers familiar with the field. From the human experiments, we proved that our model can assist astronomers' work.
2. Animal classes: The objective of classifying animals[2] was to compare the model in different fields. The labelers selected were 11 university students.

Each dataset contains 4 classes and 318 unknown objects, for about 15 people. Each user was presented

[2]The full dataset is available at: https://a-z-animals.com/animals/pictures/. We filtered the number of mammals to do not have an extremely unbalanced dataset. The class fish was removed to work with only four classes and to increase the difficulty.

with 1 to 4 random YN questions per instance. Also, the sets have (i) 40 and (ii) 41 known objects, respectively. For those known objects and 80 of the 318 unknown ones, the users were asked the ABCD question as well. The following results are based only on those labelers who finished at least 70% of the questions.

Table 1: Instances per class for each real-world dataset.

MACHO		*The Catalina Surveys*		*Animals*	
EB	104	CEP	119	Mammal	232
BE	57	RRLYR	99	Bird	73
LPB	49	EB	80	Amphibian	31
CEP	40	LPV	60	Reptile	23

7 Results

The experiments are divided into eleven parts: Two full experiments with synthetic data (7.1 and 7.2); four using classifiers on MACHO data (7.3, 7.4, 7.5, and 7.6); finally, we set up the websites to get real crowds' results, which we present in five experiments (7.7, 7.8, 7.9, 7.10, and 7.11). We used NUTS for all experiments, except for the benchmark against BBVI presented in experiment 7.7. We always used ten sampling chains and *burned* the first 1500 samples.

7.1 Convergence Simulations - Synthetic Data. We created votes, as explained in subsection 6.1. For synthetic and classifiers' votes, we used six labelers and four classes. We asked each labeler between 1 and 4 questions (Random(1,4)) for about 250 objects. Between 25 and 40 objects were used to approximate $\hat{\Theta^j}$; the rest were used for testing.

For all experiments we performed, the classification accuracy scores became completely stable after 3000 iterations. Similar results for convergence were obtained from both classifiers' scenarios and the two set-up contests with real-world data. The convergence of each variable ($\hat{\Theta}$, π, $\mathbf{Z}$, and Θ) was diagnosed based on the Gelman-Rubin statistic [8]. They all converged.

7.2 Modeling the Crowd Expertise - Synthetic Data. To prove that our model can effectively differentiate between accurate and inaccurate labelers, we compared it with the baselines used in [33]. Here, we worked with 7 synthetic labelers with higher expertise for at most two of four classes (as explained in section 6). Figure 5 shows the performance of each method after convergence. This shows that our method outperforms all the baselines when the labelers do not have equal knowledge about all classes. Since we only have YN responses, an ABCD model would not be appropriately trained.

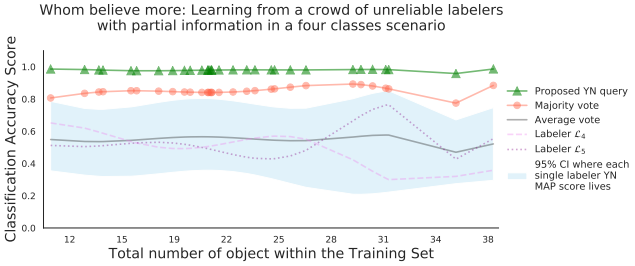

Figure 5: Crowd expertise on synthetic data. Note that each labeler score lies in a range of lower accuracy classification score than the YN and majority methods.

- *YN query*: We predicted $\mathbf{Z}$ via posterior inference.
- *Each labeler's ABCD simulated votes*: We asked one k_i^j per object to each labeler, where $k = z_i$. This means we asked if $\mathcal{X}_i$ belongs, "yes" or "no", to what we know is the true label z_i. We considered these answers as ABCD votes. We obtained the classification accuracy score as the proportion of right answers.
- *Majority vote*: As a prediction, we took the majority of the labelers' ABCD simulated votes.
- *Average vote*: Represents the average of the accuracy scores of each labeler's ABCD simulated votes.

7.3 Performance Depending on the Training Set Size - MACHO Data. First, we evaluated how many objects we would need to converge the $\hat{\Theta}$ estimation quickly. Second, we checked the model's sensitivity to the hyperparameters α and β. Figure 6 shows that the learning rate grows logarithmically with the training set size. This means that by only asking about a few known objects $\hat{\mathcal{X}}_i$, the model can quickly converge to a good estimation of $\hat{\Theta}$ and $\mathrm{P}(\mathbf{Z}|\mathbf{R})$, almost independently of $\mathcal{N}$. It also shows that this model can achieve equal results with different initial hyperparameter values.

7.4 Recovery of Credibility Matrices $\hat{\Theta}$ - MACHO Data. The accuracy classification score and the training set size are closely related, as shown in figure 6. Figure 7 shows that the convergence of $\hat{\Theta}$ also depends on the training set size. Hence, if we estimate a good $\hat{\Theta}$, we can reach a higher accuracy score. Finally, the accuracy score depends on the convergence of $\hat{\Theta}$.

7.5 Performance Simulations Depending on Θ Convergence - MACHO Data. Figure 8 shows that

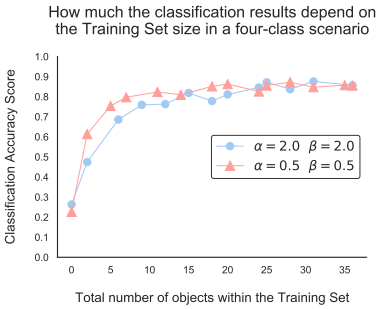

Figure 6: Classifiers voting for MACHO data. Note that increasing the training set size to about 10 instances produces an increase of 50% (from 20% to 70%) in the classification accuracy score. This shows that, after a small number of instances, the accuracy remains stable.

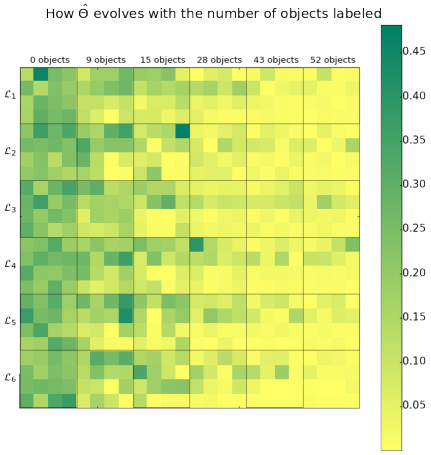

Figure 7: Classifiers voting for MACHO data MSE. MSE between each original Credibility Matrix row and its recovered $\hat{\theta}^j_{kk'}$ estimation with our method. Figures 6 and 7 both show convergence by the object number 35. If we have 36 objects and 4 classes, each labeler votes for about 9 objects per class. $\mathrm{E}\{\mathrm{Random}(1,4)\} = 2.5$ questions per object implies 22.5 questions per class, which means about 5.6 votes to estimate each $\hat{\theta}^j_{kk'}$. We can see that the convergence of $\hat{\Theta}$ depends directly on that set size.

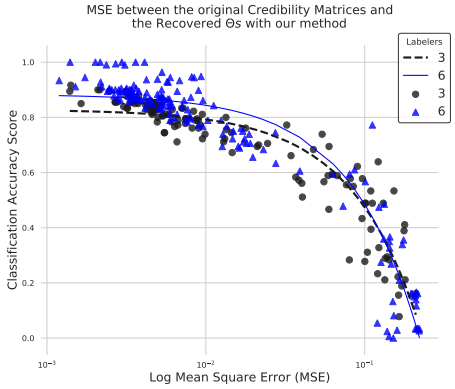

Figure 8: MSE between original Credibility Matrices and the Recovered ones. Classifiers voting for MACHO data. The error has two possible sources: i) an insufficient size of the training set, and ii) a lack of convergence in the model. In conclusion, how accurate is the estimation of $\mathrm{P}(\mathbf{Z}|\mathbf{R})$ depends on the quality of the estimation of Θ.

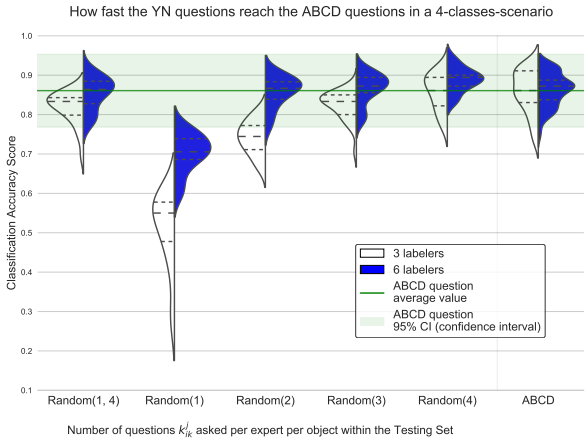

Figure 9: Classifiers voting for MACHO data. Random(w) means we asked each labeler for w different classes M_k a question k^j_{ik}, $w \leq \mathcal{K}$. The violin shape represents the cross validation results distribution.

the better the model estimates the labelers' credibilities Θ, the better the classification accuracy score.

7.6 Performance Simulations - MACHO Data. In a four-class scenario, our method reaches the performance of the ABCD method (see figure 9) when we asked Random$(1,4)$ YN queries per object per labeler. The implemented baseline is a Bayesian ABCD model, a Hybrid Confusion Matrix [17] based on DawidSkene [4] plus the prior estimation stage of confusion matrices.

In a five-class scenario, six labelers outperformed the ABCDE model when giving responses for only four classes. This means that the labelers were not required to discern among the five classes to reach high accuracy scores. However, we found that three labelers are not enough for this scenario, since they need to respond for all five classes to reach the full question model.

Scenarios with four and five classes showed that the YN model outperforms the ABCD method when we ask a YN question for every possible class $\mathcal{M}_k$ for every

object $\mathcal{X}_i$. This indicates that each YN response is more precise or confident than each ABCD response. The difference relies on the fact that in the YN model we can ask for enough explicit information to estimate each row of the credibility matrices, while in the ABDC scenario, we cannot ask queries to evaluate specific errors between pairs of classes.

7.7 Performance Real-World Votes MCMC vs. BBVI - Websites. We ran all previous simulations using the PyMC3 implementation mainly for two reasons. First, even though we used the AdaGrad [20] algorithm to set the learning rate, this setting presents more parameter tunning than does MCMC parametrization in BBVI. Second, the PyMC3 implementation usually slightly outperformed the BBVI results. Even though we also evaluated time and memory complexity, here we present only time until complete convergence.

Time Until Complete Convergence The experiments were performed for times of 10 minutes (PyMC3) versus 5 minutes (BBVI) for The Catalina Surveys full model running 1 chain; the times for the Animals Dataset were 14 minutes (PyMC3) versus 7 minutes (BBVI). Since both datasets were equal in siz, those times depend only on the number of labelers, 8 and 11 respectively for each dataset. The time spent is linear on the number of chains for both models.

Given that the experiments took minutes to converge, these implementations cannot support active learning, as each step would require converging a model to estimate the next question and labeler.

The results for The Catalina Surveys are shown in figure 10. The figure shows that for this data, the MCMC model outperforms the BBVI implementation. For the Animals Data, both implementations have a 99.7% accuracy score. The BBVI implementations are both parametrized equally. We found that the BBVI approach can get higher accuracy if we fine-tune each learning rate of the latent variables.

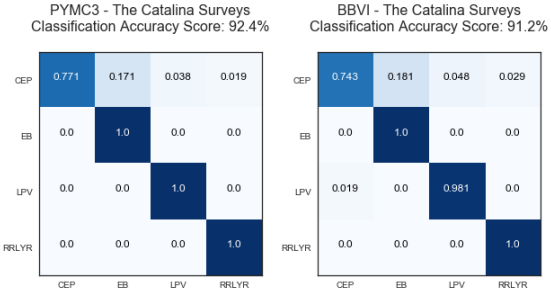

Figure 10: PyMC3 vs. BBVI. Confusion matrices for the learned models from Real-World Data.

7.8 Performance Crowd Versus Each Labeler - Websites. To evaluate the individual performance of each labeler versus the mixture of them, we trained one YN model per labeler. Figure 11 shows the three best individual performances in the The Catalina Surveys contest. The figure shows that our strategy effectively modeld and integrated the unreliable crowd knowledge.

The YN strategy can control unreliable labelers mainly for two reasons. First, the Credibility stage allows the model to discover how each labeler makes mistakes and interprets the labelers' responses. Second, the mixture of labelers helps the model to converge to a correct posterior distribution of the classes by weighting them according to their credibility matrices.

The labelers' behavior for the Animals datasets is quite similar; many of them are unreliable, but the full model is more accurate than all the labelers.

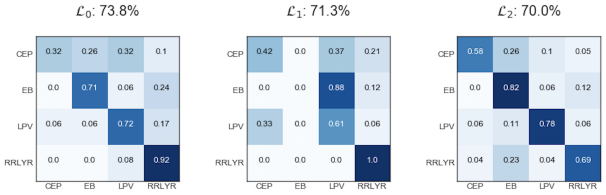

Figure 11: The Catalina Surveys contest's participants. Even though the labelers are confused, the YN model can learn how they fail. Figure 10 shows that our model outperforms each labeler.

7.9 Performance Real-World Votes YN vs. ABCD - Websites. As we explained in section 6, each labeler was presented with a series of full ABCD questions for 80 objects, for which the labelers were asked for Random$(1, 4)$ YN queries as well. For these objects, the animals contest achieved 100% accuracy with both strategies. For The Catalina Surveys, the YN query reached 91.2% and the ABCD 90.0%.

7.10 Performance Analysis YN Question vs. ABC Question - Websites. Finally, we analyzed the cost and performance of the number of YN queries versus the number of ABCD queries needed for convergence of the classification accuracy score. Although the YN query requires less expertise than the full ABCD question, the time spent on selecting an ABCD response is not proportional to the number of possible classes $\mathcal{K}$. This is shown in the websites' time records, where answering an ABCD question required less than twice the time of answering a YN question. To measure the cost, we compared how many YN queries versus how many ABCD queries are needed for the model to converge. We could assume that each ABCD query is equivalent

to give $\mathcal{K}$ YN votes [30], because each ABCD response requires the labeler to recognize the YN response for all $\mathcal{K}$ possible classes. Figure 12 shows that if 4 YN queries require as much effort as 1 ABCD question, the YN model converges faster and to a higher classification accuracy score. This occurs because the YN model can better differentiate among the possible errors, since the YN query gives specific information to estimate all the rows within the credibility matrices. As figure 8 shows, the better the model estimates the credibility matrices, the better the classification accuracy score.

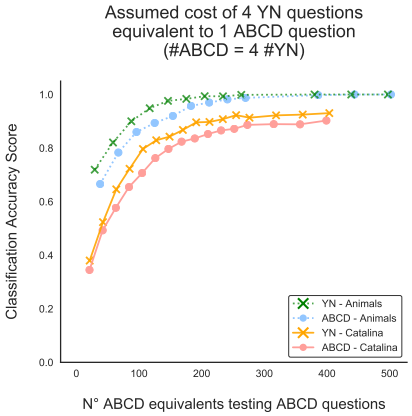

Figure 12: ABCD equivalent questions. Results from two web contests: Real-World votes on two different scenarios. The ABCD predictions were obtained from the Bayesian model described in section 7.6.

Despite assuming that 4 YN queries are equivalent to 1 ABCD query, figure 13 presents an analysis of different ABCD equivalences. All ABCD predictions were obtained from the Bayesian model described in section 7.6, which was also used in figure 9.

The analysis in figure 13 corresponds to how much difference exists between the classification accuracy score of the YN scenario and that of the ABCD scenario. The "1 ABCD = 4 YN" lines represent the differences in figure 12, where the YN surpasses the ABCD strategy. We compared this error (axis-Y) to the number of equivalent ABCD questions asked during the labeling stage (axis-X). Figure 13 illustrates that the YN strategy outperforms the ABCD strategy when we assumed that each ABCD query is equivalent to at least 3 YN queries. In addition, we can see that when asking an average of 2.5 questions per object and labeler, the YN model reached the ABCD's performance quickly. Furthermore, when we assume that each YN question is equivalent in cost to one ABCD question, at some point the YN reaches or outperforms the ABCD's performance.

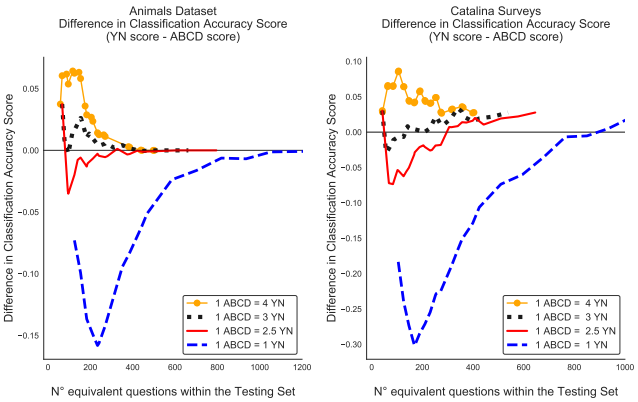

Figure 13: Difference in Classification Accuracy Scores. Results from two web contests: Real-World votes on two different scenarios. The ABCD predictions were obtained from the Bayesian model described in section 7.6.

7.11 Cognitive Cost Analysis YN Question vs. ABC Question - Websites. The amount of cognitive effort made by annotators depends on factors like the information available or the number of classes. Since we cannot evaluate all possible scenarios objectively, we show the assessment of different costs in a four-class scenario in figure 14. Figure 14 illustrates that assuming that each ABCD query is equivalent to one YN query, the model is not convenient regarding time spent. However, when the cognitive cost of a YN query is less than half that of an ABCD query, the effort made by annotators to converge the model is less than the effort required when they are asked for ABCD queries. Overall, we can see that if the cognitive cost for a YN query is less than 0.6 times that for an ABCD query, the YN strategy reduces the total effort.

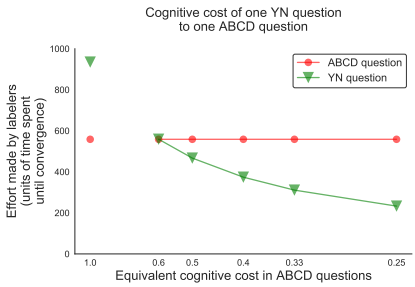

Figure 14: Results from web contests: Real-World votes on two different scenarios. ABCD predictions are from the Bayesian model described in section 7.6. The y-axis values were taken from the website scenarios. The times marked at cost 1.0 are empirical data, and any other point is proportional to the assumed cognitive effort.

8 Conclusion

We developed a new model for crowdsourcing with "yes" or "no" type queries that can be applied to any context. The YN model obtains comparable results with models that ask full questions to labelers. The reduction of labelers' efforts depends on how much cognitively easier it is to respond to a YN versus an ABCD question. Furthermore, our model convergences more quickly without sacrificing accuracy. We could also see that in cases where most labelers are unreliable, the YN model was able to capture the right posterior of the classes by taking advantage of crowds.

As a future work, the model could capture variations in expertise over time. Also, here we randomly selected an object along with a class; this election could be optimized using an active learning approach or by understanding the biases produced by the order in which the pairs of objects and questions are presented to the labelers.

Acknowledgements

Our work was supported in part by the CSS survey, which is funded by the National Aeronautics and Space Administration under Grant No. NNG05GF22G issued through the Science Mission Directorate Near-Earth Objects Observations Program. We would also like to thank the anonymous reviewers whose comments greatly improved this manuscript.

References

[1] Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory, ACM, pp 92–100

[2] Chaney AJ (2015) A guide to black box variational inference for gamma distributions

[3] Cook KH, Alcock C, Allsman R, Axelrod T, Freeman K, Peterson B, Quinn P, Rodgers A, Bennett D, Reimann J, et al (1995) Variable stars in the macho collaboration 1 database. In: International Astronomical Union Colloquium, Cambridge University Press, vol 155, pp 221–231

[4] Dawid AP, Skene AM (1979) Maximum likelihood estimation of observer error-rates using the em algorithm. Applied statistics pp 20–28

[5] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, pp 248–255

[6] Drake A, Djorgovski S, Mahabal A, Beshore E, Larson S, Graham M, Williams R, Christensen E, Catelan M, Boattini A, et al (2009) First results from the catalina real-time transient survey. The Astrophysical Journal 696(1):870

[7] Gelfand AE, Smith AF (1990) Sampling-based approaches to calculating marginal densities. Journal of the American statistical association 85(410):398–409

[8] Gelman A, Rubin DB (1992) Inference from iterative simulation using multiple sequences. Statistical science pp 457–472

[9] Geman S, Geman D (1984) Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on pattern analysis and machine intelligence (6):721–741

[10] Hastings WK (1970) Monte carlo sampling methods using markov chains and their applications. Biometrika 57(1):97–109

[11] Hoffman MD, Gelman A (2014) The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. Journal of Machine Learning Research 15(1):1593–1623

[12] Huang SJ, Chen S, Zhou ZH (2015) Multi-label active learning: Query type matters. In: IJCAI, pp 946–952

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

[14] Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT press

[15] Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

[16] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural computation 1(4):541–551

[17] Liu C, Wang YM (2012) Truelabel+ confusions: A spectrum of probabilistic models in analyzing multiple ratings. arXiv preprint arXiv:12064606

[18] Pichara K, Protopapas P, Leon D (2016) Meta-classification for variable stars. The Astrophysical Journal 819(1)

[19] Qi GJ, Hua XS, Rui Y, Tang J, Zhang HJ (2008) Two-dimensional active learning for image classification. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, pp 1–8

[20] Ranganath R, Gerrish S, Blei DM (2014) Black box variational inference. In: AISTATS, pp 814–822

[21] Rashidi P, Cook DJ (2011) Ask me better questions: active learning queries based on rule induction. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 904–912

[22] Ratner AJ, De Sa CM, Wu S, Selsam D, Ré C (2016) Data programming: Creating large training sets, quickly. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R (eds) Advances in Neural Information Processing Systems 29, Curran Associates, Inc., pp 3567–3575, http://papers.nips.cc/paper/6523-data-programming-creating-large-training-sets-quickly.pdf

[23] Raykar VC, Yu S, Zhao LH, Valadez GH, Florin C, Bogoni L, Moy L (2010) Learning from crowds. Journal of Machine Learning Research 11(Apr):1297–1322

[24] Salvatier J, Wiecki TV, Fonnesbeck C (2016) Probabilistic programming in python using pymc3. PeerJ Computer Science 2:e55

[25] Schapire RE, Freund Y (2012) Boosting: Foundations and algorithms. MIT press

[26] Shah, N. B., Zhou, D. (2015) Double or nothing: Multiplicative incentive mechanisms for crowdsourcing. In: Advances in Neural Information Processing Systems, pp. 1–9.

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

[28] Vaughan, J. W. (2018). Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research. Journal of Machine Learning Research, 18(193), pp. 1–46.

[29] Wainwright MJ, Jordan MI, et al (2008) Graphical models, exponential families, and variational inference. Foundations and Trends® in Machine Learning 1(1–2), pp. 1–305

[30] Welinder P, Perona P (2010) Online crowdsourcing: rating annotators and obtaining cost-effective labels. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, IEEE, pp 25–32

[31] Yan Y, Fung GM, Rosales R, Dy JG (2011) Active learning from crowds. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 1161–1168

[32] Yan Y, Rosales R, Fung G, Dy J (2012) Modeling multiple annotator expertise in the semi-supervised learning scenario. arXiv preprint arXiv:12033529

[33] Yan Y, Rosales R, Fung G, Schmidt MW, Valadez GH, Bogoni L, Moy L, Dy JG (2010) Modeling annotator expertise: Learning when everybody knows a bit of something. In: International conference on artificial intelligence and statistics, pp 932–939

[34] Zhang C, Chaudhuri K (2015) Active learning from weak and strong labelers. In: Advances in Neural Information Processing Systems, pp 703–711