Noname manuscript No.
(will be inserted by the editor)

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

Belén Saldías-Fuentes · Pavlos Protopapas · Karim Pichara

Received: date / Accepted: date

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

Keywords supervised learning, crowdsourcing, query type, training sets, variational inference, classification

Belén Saldías-Fuentes · Karim Pichara
Computer Science Department, Pontificia Universidad Católica de Chile, Santiago, Chile
Tel. Saldías F.: +569 98435827
E-mail: bcsaldias@uc.cl; kpb@ing.puc.cl

Pavlos Protopapas
Institute for Applied Computational Science, Harvard University, Cambridge, MA, USA
E-mail: pavlos@seas.harvard.edu

[1] Available at: the link will be revealed at soon as the paper gets published.

1 Introduction

Labeled data is the very first requirement for training machines. In classification tasks, models need to be taught about which objects belong to which classes. The issue is that unlabeled data increases much faster than labeled data. Moreover, the availability of data has stimulated great breakthroughs in AI, more than the latest models have done. For example, convolutional neural networks (CNNs) were first proposed by LeCun et al (1989), but only when ImageNet (Deng et al, 2009) achieved a corpus of 1.5 million labeled images and 1,000 object categories allowed Google's GoogLeNet (Krizhevsky et al, 2012; Russakovsky et al, 2014) to perform object classification almost as good as humans by using CNNs. This encourages us to focus on creating new mechanisms for producing labels. Nevertheless, labeling means getting ground truths, which is in most cases hard, expensive, or impossible to get.

In order to increase the amount of labeled data, we can use crowdsourcing (Dawid and Skene, 1979; Raykar et al, 2010; Simpson et al, 2013; Vondrick et al, 2013). In the absence of a perfect oracle, we can take advantage of crowd's knowledge to create training sets and classify, even though they may give imperfect opinions. Working with crowds of labelers requires less effort from each labeler since each one does not need to tag all the data. A big challenge is to combine unreliable crowd information: not entirely accurate, but cheaper (Yan et al, 2012). A typical case dealing with crowds is to trust each labeler equally. For example, taking the simple majority of votes for each object. For this to work, we must assume everyone knows the same about the topic. This is in many cases a wrong assumption. For instance, in an observatory with two senior astronomers and four new interns, the astronomers propose their best telescopes setting, but the interns reject to that configuration because they may not understand all the implications. We cannot just trust the majority.

From a different strategy, creating training sets has been approached with active learning (AL) solutions (Settles, 2010; Zhang and Chaudhuri, 2015; Yan et al, 2011). AL is a semi-supervised scenario in which a learning algorithm, iteratively selects the best instances to ask a user for labels (for example the objects that confuse most the model). AL assumes a user as perfect labeler, whose responses are used to retrain the model.

Several different solutions for crowdsourcing with active learning scenarios have been proposed (Simpson et al, 2013; Zhang and Chaudhuri, 2015; Vondrick et al, 2013; Liu and Wang, 2012; Yan et al, 2012; Raykar et al, 2009) (see section 2). Most of them approach the solution by selecting the best instances as candidates to be tagged, considering the labelers' expertise and the way to combine the crowd's answers. Nevertheless, the main challenge remains on that manual labeling is extremely difficult when unlabeled data is abundant.

In this paper, instead of selecting the best instances as candidate labels for training the model, which as described above it is the traditional approach to crowdsourcing and active learning, we propose a novel approach based on the query type. Commonly in a four classes scenario, a labeler is asked the class of an object, with possible responses "A" or "B" or "C" or "D". Under this settings, we refer to the full question as ABCD question. The proposed model generates low-cost queries where each response produces partial information. This method iteratively selects, per labeler, a

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

Fig. 1: Different query scenarios. Note that for the YN question the labelers do not need to know what is the ground truth to give accurate partial information. The 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 showed animal.

random object along with a random class' label, and then asks if that object belongs "yes" or "no" to that class (proposed YN question). See figure 1 for an example.

To represent the labelers' errors per response in a full ABCD question scenario, traditional models use different confusion matrices (Liu and Wang, 2012; Simpson et al, 2013), where each row can be modeled by a Dirichlet distribution. In our context, given a multi-class scenario, we need to measure the probability per labeler of giving a binary right answer, when the class asked is $\mathcal{M}_{k'}$ and the true class is $\mathcal{M}_k$. We represent those probabilities in what we call a *credibility matrix* per labeler, where each cell in a row can be modeled by a Beta distribution. Furthermore, we introduce the idea and proof the importance of having a supervised stage to learn the YN model. In a first stage, we ask for few known object labels with the aim of estimating prior credibility matrices. As we show in section 7, using that information we can ensure convergence, other way makes the inference model hard or impossible to converge to the true labels. The second stage is the labeling of unknown objects.

The proposed method has many advantages over traditional approaches. First, the YN model focuses on the importance of learning a prior estimation of how the labelers fail. The YN strategy probabilistically learns this estimation as a prior parameters estimation for the labeling stage. Second, it provides partial information with less error. Third, the labelers do not need to know about all the classes. Finally, the method is independent of the kind of data, given that we only need to include labelers' votes, without worrying about any 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 of getting data reduces the cost of other models because it can reconstruct the ground truth labels dealing only with partial information. We show that the aggregation of partial information allows the YN model to ask fewer questions than others, for getting convergence.
2. *New data released*: We develop two real-world experiments with human crowds and publish the data.

The rest of this paper is organized as follow: In section 2 we present a brief description of the related work in crowdsourcing with imperfect labelers and active learning approaches. Section 3 explains the proposed model, and section 4 presents the approached inference scheme to solve it. Then, in section 5 we describe our implementations of the model. Section 6 describes the different dataset used for the experimental results exposed in section 7. Then, section 7 shows extensive experiments and analysis. Finally, in section 8 we discuss and conclude the main results of our work.

2 Related Work

We discuss three relevant areas of related work in this section. First, the importance of training sets. Second, crowdsourcing with active learning for classification scenarios. Third, approximate variational inference approaches to crowdsourcing models.

2.1 Creating Training Sets

Creating training sets has become a central topic of supervised learning and generative models. Scientific works show that a model is as good as its training set[2]. For creating labels, we can manually label as many objects as we can. Furthermore, we can distribute the work among a crowd of labelers who gives labels. An independent approach is to use active learning (AL) (Settles, 2010). AL assumes the presence of a perfect oracle, which gives labels only for the objects that improve most the model performance. We explore deeply crowdsourcing and active learning in the subsection 2.2 and 2.3.

From another point of view, it is possible to use a programmatic paradigm to create training sets, which is called *data programming* (Ratner et al, 2016). In this paradigm, labelers give no labels but heuristics or strategies as labeling functions that return the asked labels. Furthermore, this approach is extended to crowdsourcing scenarios. There is a big difference between the proposed YN model and *data programming*, starting with the query type. *Data programming* asks for a function while we ask only for "yes" or "no" responses. An additional mechanism to label data is *co-training* (Blum and Mitchell, 1998), where the unlabeled data is classified under two conditional independent views. Each view is trained with a small amount of labeled data. In comparison to *co-training*, we can handle as many views as we want.

[2] http://www.kdnuggets.com/2016/05/datasets-over-algorithms.html

Finally, close to our approach is the semi-supervised *boosting* (Schapire and Freund, 2012) procedure, which combines the output of several "weak" classifiers to create a "strong" classifier. In this case, we approach the weaknesses by modeling the labelers errors to infer the true labels probabilistically.

2.2 Crowdourcing Scenarios

There are several approaches to crowdsourcing with active learning that use the most relevant instances for being labeled by unreliable or noisy labelers (Simpson et al, 2013; Zhang and Chaudhuri, 2015; Vondrick et al, 2013). They all try to make the labeling task easier. Some works propose probabilistic graphical models (PGM) (Koller and Friedman, 2009; Wainwright et al, 2008) as well as we do (Liu and Wang, 2012; Yan et al, 2012). However, we differ from them mainly on that they require the labelers to give the class as a full ABCD answer while in our work we can handle partial information asking for YN questions. In the YN scenario, labelers do not need to discern among all the possible classes. Furthermore, their selection criterion focuses the analysis by either selecting the instances to be labeled or asking the best labeler for classification (Raykar et al, 2009). We address the creation of training sets and labeling to a new goal, a new query type that diminishes the effort required from the labelers. In addition, several efforts have been made on how to estimate the labelers' expertises (Yan et al, 2010,0; Liu and Wang, 2012; Simpson et al, 2013; Zhang et al, 2014). We propose to estimate those expertises as prior of the inference model. Results show that using this semi-supervised approach the proposed model can converge quickly by asking only for few training objects.

There are some works that propose new query types on active learning scenarios. For instance, a different approach to ours (Rashidi and Cook, 2011) reduces the number of questions by asking generic queries. These queries are generated by rule induction from multiple unlabeled instances. Another approach classifies images by asking a different type or queries (Huang et al, 2015). This approach relies on asking about what label (given a pair of labels) is more relevant for each instance, where each image can be classified into more than one category. The closest research to the YN query type (Qi et al, 2008) assumes each instance could belong to more than one class. This assumption allows the model to take into account the label correlation between the classes for the classification. The main difference between the YN model and all these works relies on the fact that they do not propose a crowdsourcing context to improve the scenario. They keep mainly the existence of a perfect oracle assumption. A work that changes this perfect oracle and the query type is closer to the hierarchical crowdsourcing (Otani et al, 2015), but it still asks full questions.

Until now, no paper has been presented for the integration of the query type importance, partial information requested to the labelers, and the crowd's power. We propose a mechanism that outperforms and handles many difficulties, as we expose in section 1 and through this work.

2.3 Variational Inference Approaches

The crowdsourcing strategies discussed in 2.2, along with the proposed YN method, can be solved using several inference schemes. In this subsection, we present some of these schemes and address reasons to use approximate inference for the YN strategy.

The Bayesian model and techniques proposed by Blei et al (2003) pushed up several works based on probabilistic graphical models and variational inference (VI) (Jordan et al, 1999; Wainwright et al, 2008) methods. In this area, works such as Raykar et al (2010); Yan et al (2012) solve the semi-supervised learning scenario from a probabilistic perspective using mainly EM and MAP algorithms. However, since the YN model involves high-dimensionality estimation (due to the credibility matrices estimation, see subsection 3.3), it is very likely for EM and MAP to converge to a local optimum.

Works that handle high-dimensional estimations use mainly the Gibbs sampler (Geman and Geman, 1984) to infer ground truth labels for objects in crowdsourcing scenarios (Liu and Wang, 2012; Carpenter, 2011). There is also a nonparametric approach to labels inference that uses Gibb sampling (Moreno et al, 2015). These techniques are the closest to the proposed Markov chain Monte Carlo (MCMC) (Hastings, 1970; Gelfand and Smith, 1990) scheme for the YN model. We address the sampling using the No-U-Turn Hamiltonian sampler (NUTS) (Hoffman and Gelman, 2014) to converge faster than using a random walk as simple MCMC does.

Additionally, Simpson et al (2013) propose a full variational treatment for crowdsourcing scenarios. In this last work, they provide all the mean field equations to approximate the target distribution and evaluate the lower bound. Regarding to this variational approach, we propose a second method to solve the YN model, because variational inference usually is faster than MCMC and NUTS. Hence, we find a simple approximate solution using Black Box (Ranganath et al, 2014) variational inference method. This method is inexpensive and easy to implement because it maximizes the ELBO by only estimating its gradient.

Following the full variational approach, the crowdsourcing scenario with unreliable labelers can be transformed into a standard inference problem in graphical models (Liu et al, 2012). This approach proposes to use belief propagation and the mean field theory connected to EM. The performance of this method critically depends on the choice of a prior distribution on the labelers' confusions. We propose an automatic solution for this problem, to separate the method into two stages: the first stage estimates how reliable the labelers are by asking for known objects labels (Bragg et al, 2016), and the second stage asks for unknown objects labels. In the full-question-scenario, Wauthier and Jordan (2011) propose to unify these stages in one Bayesian model that captures the labelers bias through latent variables. However, they still work with full information and the YN model works only with partial information.

All works proposed in this section have always studied either the method for a two-coin model or a multi-class scenario where they ask for full questions about what is the class, never changing the question asked the labelers as we do.

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 the 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 the 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 every response r_{ik}^j. Note that classes are not necessarily in equal proportions, every labeler is not necessarily asked for all objects, and a labeler is not asked twice for the same class for the same object.

We propose a probabilistic graphical model to infer the true labels $\mathbf{Z}$, this is shown in figure 4. The `Labeling` area represents the joint distribution of the true labels and the rest of the variables involved in their prediction. These variable distributions are assumed as it is presented in this section.

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

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

Fig. 2: Responses/votes r_i^j.

3.2 Credibility Matrices

Common Bayesian approaches for crowdsourcing classification use the confusion matrix of each labeler to represent their errors, due to the nature of the full question. We represent the YN error per labeler as a *credibility matrix*. In our context, we need to measure 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 presents the credibility matrix of a specific labeler, where $\theta^j_{kk'}$ represents the probability of the labeler $\mathcal{L}_j$ of saying "yes" to the question k'^j_i when $z_i = k$. We assume that the labelers are not random voters so that we can find patterns in their behaviors.

The main goal of our model is to decide the most likely class for each object, given the votes of different labelers and their *credibility matrices* Θ. A side goal is to estimate those credibility matrices. In particular, we consider conjugate priors. Given that each "yes" or "no" response can be modeled as a $\mathrm{Bernoulli}$ distribution, the prior for each $\theta^j_{kk'}$ distributes as (1), where $\hat{\alpha}^j_{kk'}$ and $\hat{\beta}^j_{kk'}$ are the estimated prior initial parameters (see section 4).

$$\theta^j_{kk'} \sim \mathrm{Beta}(\hat{\alpha}^j_{kk'}, \hat{\beta}^j_{kk'}) \tag{1}$$

Finally, the likelihood is:

$$r^j_{kk'} \sim \mathrm{Bernoulli}(\theta^j_{kk'})$$

Modeling the prior of $\theta^j_{kk'}$ as a Beta distribution that lives in a 0 to 1 space allows 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.

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

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

3.3 Joint Distribution

Each YN vote r^j_{ik} depends on the real, but unknown label z_i. Furthermore, the vote also depends on the credibility $\theta^j_{z_i k}$ of the 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 allows us to model the labeler's biases and errors for all the classes. These dependencies are represented by the conditional distribution $\mathrm{P}(r^j_{ik}|z_i, \theta^j_{z_i k})$ (Liu and Wang, 2012).

From prior information, we can estimate the initial classes proportions ρ, and we can define a global Dirichlet variable π in charge of this unknown distribution of the vector $\mathbf{Z}$. The distribution that governs this random variable is:

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

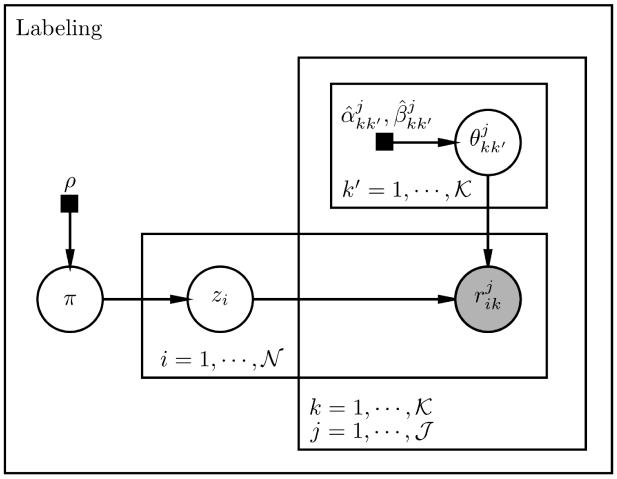


Fig. 4: Proposed PGM. In this plate notation, random variables are clear circles; observed variables are shaded in gray. Estimated prior hyperparameters are represented by squares.

The distribution for each z_i is:

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

Likelihood

We start from a single labeler, one object, and one question. For labeler $\mathcal{L}_j$ and question k^j_j the likelihood is in (2).

$$\mathrm{P}(r^j_{ik}|\theta^j_{z_i k}, z_i) = \overbrace{(\theta^j_{z_i k})^{\overbrace{r^j_{ik}[0]=1, r^j_{ik}[1]=0}^{YES}}}^{r^j_{ik}[0]} \times \overbrace{(1-\theta^j_{z_i k})^{\overbrace{r^j_{ik}[0]=0, r^j_{ik}[1]=1}^{NO}}}^{r^j_{ik}[1]} \tag{2}$$

For all responses $\mathbf{R}$, all labelers $\mathcal{L}$, and all data $\mathcal{N}$ the likelihood is in (3).

$$\mathrm{P}(\mathbf{R}|\Theta, \mathbf{Z}) \propto \prod_{i=1}^{\mathcal{N}} \prod_{j=1}^{\mathcal{J}} \prod_{k \in \mathcal{K}^j_i} \left\{ (\theta^j_{z_i k})^{r^j_{ik}[0]} \, (1-\theta^j_{z_i k})^{r^j_{ik}[1]} \right\} \tag{3}$$

4 Inference Schema

In the crowdsourcing problem we address in this work, every instance in the dataset could be voted by no, one, or many unreliable labelers. In consequence, we need to be aware of the labeler's credibilities. The approached solution is to ask them for a

training set to estimate their initial credibilities. In this credibility estimation stage, the main issue is how much to ask each labeler.

To train the YN graphical model, we use a first supervised stage. Here we show examples to the labelers to obtain an initial parameters estimation; we call this process the prior `Credibility` estimation. Following this idea, in this stage, the users are presented with a set of $\hat{\mathcal{N}}$ objects $\hat{\mathcal{X}}_i$ that we know the true labels $\hat{\mathbf{Z}}$ (we refer to this set as Training Set). In this scenario, where $\hat{\mathbf{Z}}$ are observed values, the model can learn beforehand an estimation $\hat{\Theta}$ and converge faster, as we show in section 7. Therefore the likelihood for all responses $\hat{\mathbf{R}}$, all labelers $\mathcal{L}$, and all data $\hat{\mathcal{N}}$ is in (4).

$$\mathrm{P}(\hat{\mathbf{R}}, \hat{\mathbf{Z}} \,|\hat{\Theta}) \propto \prod_{i=1}^{\hat{\mathcal{N}}} \prod_{j=1}^{\mathcal{J}} \prod_{k \in \hat{\mathcal{K}}_i^j} \left\{ (\hat{\theta}^j_{\hat{z}_i k})^{\hat{r}^j_{ik}[0]} \, (1 - \hat{\theta}^j_{\hat{z}_i k})^{\hat{r}^j_{ik}[1]} \right\} \tag{4}$$

The prior distribution of each $\hat{\theta}$ is chosen to be uninformative, but flexible enough to represent either a labeler with high or low expertise. We select $\mathrm{Beta}(\alpha, \beta)$ distributions with expected value equivalent to 0.5. We test using $\alpha = \beta = 0.5$ and $\alpha = \beta = 2$ (see section 7). Therefore,

$$\hat{\theta}^j_{kk'} \sim \mathrm{Beta}(\alpha, \beta)$$

Hence, this inference scheme works in two stages as it is shown in figure 5:

1. `Credibility` stage: estimating the prior credibility $\hat{\Theta}$. Because we assume the labelers will behave similarly in the `Labeling` stage as they do in the this stage, here we obtain the $\hat{\alpha}^j_{kk'}$ and $\hat{\beta}^j_{kk'}$ parameters (equation 1) that come from each $\hat{\theta}^j_{kk'}$.
2. `Labeling` stage: running the classification labeling model to infer $\mathbf{Z}$ and Θ. We predict the labels $\mathbf{Z}$ via posterior inference.

Even though when the prior credibility estimation can be done analytically by updating the $\hat{\alpha}^j_{kk'}$ and $\hat{\beta}^j_{kk'}$ parameters, we propose to learn those parameters using posterior inference in each stage.

5 Implementation

The two-stages-model was solved using two different implementations in Python 3.5 (van Rossum and , eds): Monte Carlo Sampling and Black Box Variational Inference (Ranganath et al, 2014). Each implementation works as follow: First, it estimates the latent variables $\hat{\Theta}$. Second, it estimates Θ, $\mathbf{Z}$, and π.

5.1 Monte Carlo Sampling - MCMC

We used MCMC sampling from PyMC3 (Salvatier et al, 2016), that uses NUTS. This implementation includes mainly a DensityDist PyMC3 variable to evaluate the log-likelihood. All the experiments presented in section 7 were run using the PyMC3 implementation, except when indicated otherwise.

5.2 Variational Inference - BBVI

Due to the convergence time of the PyMC3 implementation, we decided to compare it with BBVI (Ranganath et al, 2014). This approach approximately tries to find a simple probability distribution that is closest (in KL divergence) to the true posterior distribution. We provide the derivation of the needed gradients to solve the model in appendix B (based on Chaney (2015)). These equations can be easily extended to any model with similar variable types.

6 Data

To compare the performance of our proposed model to other methods and baselines, we use simulated votes and real-world datasets with human crowds. First, we simulated data to understand the YN model 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 the labelers and their votes we proceeded as follows: First, we created labels ($\hat{\mathbf{Z}}$ and $\mathbf{Z}$). Then, for each labeler, we created a credibility matrix. The modeled labelers have high expertise in at most half of the classes. Therefore each row is simulated using a $\mathrm{Beta}(0.5, 0.5)$ distribution except where the labeler is more accurate; those are with $\mathrm{Beta}(20, 1)$ distributions (because its expected value is close to 1). Finally, we simulate the votes using the labelers and true labels. When the labeler

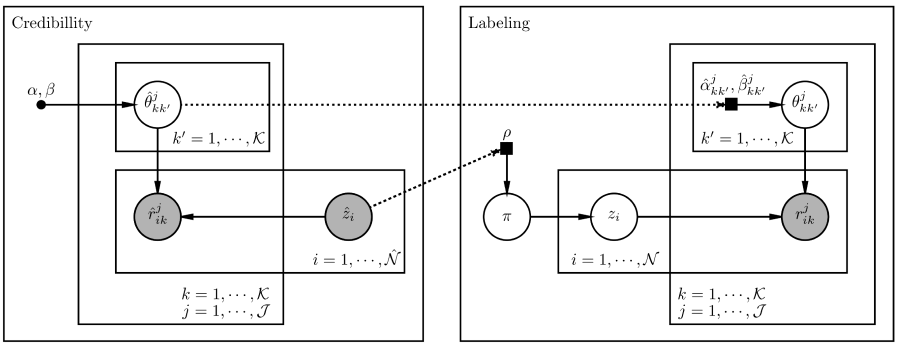


Fig. 5: Proposed inference scheme for the YN model. Dashed lines represent prior parameters estimation in the `Credibility` stage. These parameters are then used as input to the `Labeling` stage.

$\mathcal{L}_j$ is presented with the object $\mathcal{X}_i$ of class z_i, and it is asked k_i^j, we go to its credibility matrix to obtain the response for k_i^j. We take r_{ik}^j by flipping a coin with the probability given by $\theta_{z_i k}^j$.

6.2 Synthetic Votes for Real-World Data

The real dataset is a subset of the astronomical MACHO data (Cook et al, 1995) (we used 250 objects of 65 features). 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 proceed as follow: 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 is composed of a pool of $\mathcal{K}$ one-vs-all classifiers, one per class. When a labeler is asked for k_i^j, we go to its one-vs-all binary classifier for the class $\mathcal{M}_k$ to get the probability of the object to belong to the class M_k. Then we flip a coin with that probability to obtain $\hat{\mathbf{R}}$ and $\mathbf{R}$.

MACHO data: The Massive Compact Halo Objects (MACHO) project provides a time series (light curves) dataset. The main purpose this project is the detection of microlensing events in the Magellanic Clouds and Galactic bulge. Table 1 shows the distribution of the subset used in our experiments. The four classes selected are: Eclipsing Binary stars (EB), Be stars (BE), Long period variable stars (LPV), and Cepheid stars (CEP). Random sampling was used to divide the selected dataset into three different sets: i) for training classifiers, ii) for training the credibility stage, and iii) for testing the final classification. The main challenges dealing with this data relies on the fact that those time series are not uniformly sampled. In addition, they do not have a defined shape or structure to be easily handled by every model. Several works have put their efforts on classifying astronomical irregular time series (Pichara and Soto, 2011; Pichara and Protopapas, 2013; Pichara et al, 2016). Proposing a low-cost model to classify this type of data extends to the astronomers a strategy for executing their classification tasks faster.

Table 1 Real datasets classes distributions

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

6.3 Real Votes for Real-World Data

Two websites[3] were set up to acquire data from human crowds. Each of them presented a contest to people related to the dataset domain. The domains were:

1. Astronomical irregular time series: The relevance of irregular time series encourages us to handle their challenges in classification. From the human experiments, we can show empirical evidence of how our model can assist the astronomers' work. The first contest was to classify images of irregular time series of the Catalina Surveys (Drake et al, 2009). This surveys is composed first by the Catalina Sky Survey (CSS), which involves searches for rapidly moving Near Earth Objects (NEOs), and second by the Catalina Real-Time Transient Survey (CRTS), which includes searches for stationary optical transients (OTs). The labelers were astronomers and engineers related to the field. The experiments were done with a total of 8 labelers. Table 1 shows the selected subset of the Catalina Surveys.
2. Animals classes: Each labeler was presented with several images of each object. The objective of classifying animals was to compare the model in different fields. In section 7, we show that even though the presence of unreliable labelers, the YN model can reach high accuracy score. Table 1 shows the characteristics of this dataset[4]. The labelers selected were 11 university students.

Each dataset contains 4 classes and 318 unknown objects, for about 15 people. Each user was presented with 1 to 4 random YN questions per instance. Also, the sets have (i) 40 and (ii) 41 known objects. For those known objects and 80 of the 318 unknown objects, the users were asked the full ABCD questions as well, which ask for what is the class. The following results are based only on those labelers who finished at least 70% of the questions.

7 Results

In this section, we develop several experiments to evaluate the YN model performance. Our experiments were divided into eleven parts: First, two full experiments with synthetic data (7.1, 7.2). Then, four aspects using classifiers as labelers on MACHO data (7.3, 7.4, 7.5, and 7.6), and finally, we set up the websites to get real crowd's results, which we present in five experiments (7.7, 7.8, 7.9, 7.10, and 7.11). All the results are with the MCMC implementation, except a benchmark against BBVI presented in experiment 7.7. We applied ten sampling chains per experiment to validate our results.

We start checking for convergence (7.1) to be able to apply the model to different scenarios, not only to the synthetic ones. Then we set up a full synthetic scenario to model noisy crowds' expertise (7.2); we want to simulate a real context before going

[3] Available at: the link will be revealed at soon as the paper gets published.

[4] The full dataset is available at: https://a-z-animals.com/animals/pictures/. We filter the data mainly in the number of mammals to make the contests commensurable. The class fish was removed to work with only four classes, and to increase the difficulties as well.

for human votes to understand the YN model behavior. In this experiment, we want to ensure the YN method can differentiate whom to trust more and whom to ignore.

After that, we start with the classifiers' simulated votes on MACHO data. We need to decide how much to ask the labelers (7.3) to invest most of their time in labeling unknown objects. We also check for how fast the proposed model can learn the labelers' errors depending on the training set size (7.4), and how this learning rate affects the final classification accuracy score (7.5). Furthermore, using the classifiers' votes, we approximate the number of questions needed to recover as much information as the full ABCD question does (7.6).

Finally, with the real-world votes from the websites, we compare the two implementations that we develop (7.7). We evaluate classification performance, implementation technique, and memory and time spent to converge. Following that, we want to measure how well our model performs working with a crowd in the two different real-world scenarios. Therefore, we compare the YN full model results against what we get by running the model with only one labeler, where we have much less available votes (7.8, 7.9). For both real-world scenarios we define different ABCD equivalent questions. We use equivalent questions to evaluate how many YN queries we need to achieve comparable results with the full ABCD query type (7.10), and to discuss cognitive cost (7.11).

7.1 Convergence Simulations on Synthetic Data

Given that the goal of the YN model is to perform inference for the unknown variables $\hat{\Theta}$, π, $\mathbf{Z}$ and Θ, in this subsection, we check for convergence of all of these variables and accuracy score as well.

To check for convergence, we generate votes as we explained in subsection 6.1. For synthetic and classifiers' votes, we work with six labelers and four classes. We ask each labeler a random number of questions for about 250 objects in each case. We ask from 1 to 4 (Random(1,4)) YN queries per instance to each labeler. Between 25 and 40 instances were used to approximate each credibility matrix $\hat{\Theta}^j$, the rest was used for testing.

7.1.1 Classification accuracy score and labeling convergence

For all the experiments we perform, the classification accuracy score becomes completely stable after 3000 iterations. For every analysis done we *burn* the first 1500 samples, *thinning* the chains each 10 samples. Similar results for convergence are obtained from both classifiers' scenarios, and the two set up contests with real-world data.

In addition, to check for convergence of the full model, we analyze each variable convergence. The convergence diagnostics for our random variables is based on Gelman-Rubin statistic (Gelman and Rubin, 1992). To try this diagnostic, we need multiple chains to compare the similarity between them. Our experiments are based on 10 chains each. When the Gelman-Rubin ratio (potential scale reduction factor) is less than 1.1, it is possible to conclude that the estimation has converged. Figure 6

presents the potential scale reduction factors for all the estimated variables. According to this figure, there is no disagreement on that each z_i converges.

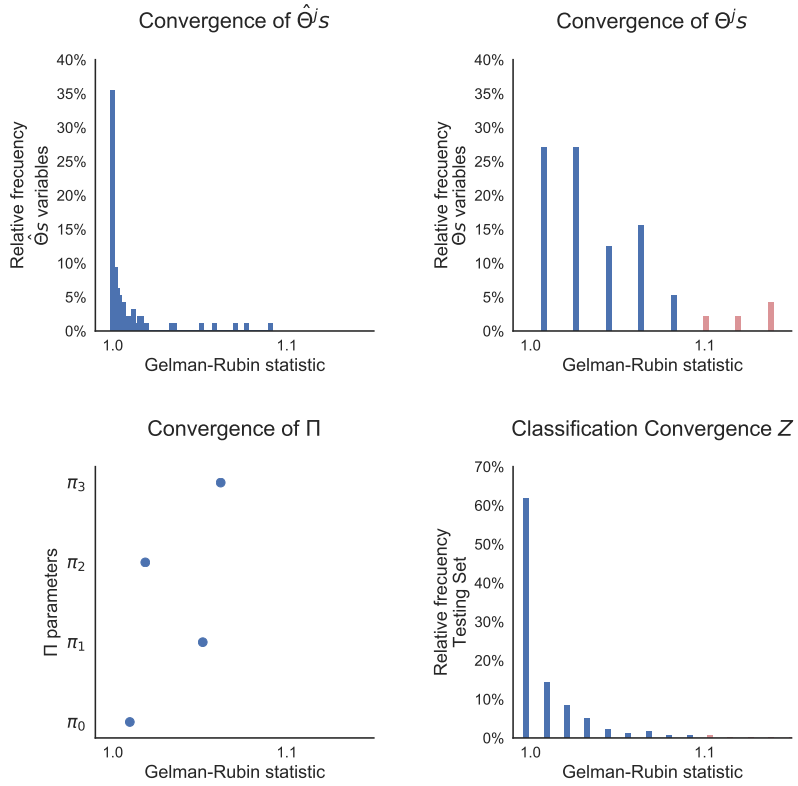


Fig. 6: Accuracy score convergence. It is possible to see that Θ has more variance than any other variable, then some of the θs have not completely converged. We cannot conclude that the model has not converged, we only can say one of the chains has not converged. In practice that minimum percentage is not conditioning the full model.

7.2 Modeling the Crowd Expertise on Synthetic Data

To prove that our model can effectively differentiate between accurate and inaccurate labelers, we propose the comparison baselines as the ones used in Yan et al (2010). In this scenario, we work with 7 synthetic labelers with higher expertise for at most two of four classes (as explained in section 6).

– *Proposed YN query*: We predict the labels $\mathbf{Z}$ via posterior inference.
– *Each labeler's ABCD simulated votes*: We ask one YN question k_i^j per object to each labeler, where $k = z_i$. It means we ask if $\mathcal{X}_i$ belongs "yes" or "no" (YN questions), to what we know is the true label z_i. Per labeler, we consider these answers as the full ABCD votes. We get the classification accuracy score as the proportion of right answers.
– *Majority vote*: As the prediction, we take the majority of the labelers' ABCD simulated votes.
– *Average vote*: This baseline represents the average of the accuracy scores of each labeler's ABCD simulated votes.

Figure 7 shows the performance of each method after convergence. We can see how our method outperforms all the baselines when the labelers do not have equal knowledge about all the classes.

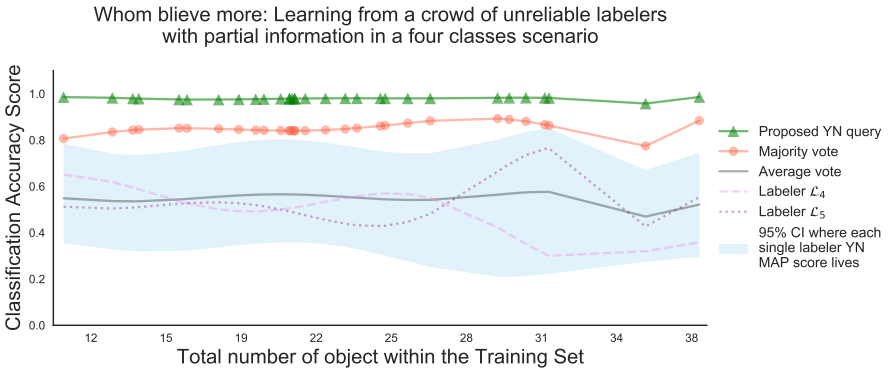


Fig. 7: Modeling the crowd expertise on synthetic data. Note that each labeler score lives in a range of lower accuracy classification score than the YN and majority methods.

7.3 Performance Depending on the Training Set Size on MACHO Data

Firstly, we want to know how many objects we need, to converge the $\hat{\Theta}$ parameters estimation quickly. Secondly, we want to check how sensitive is our model to the hyperparameters α and β. Figure 8 shows that the learning rate growths logarithmically on the training set size. It means that by asking just for 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 on the testing set size $\mathcal{N}$. In addition, it shows that the YN model can achieve equal results while using different initial hyperparameter values.

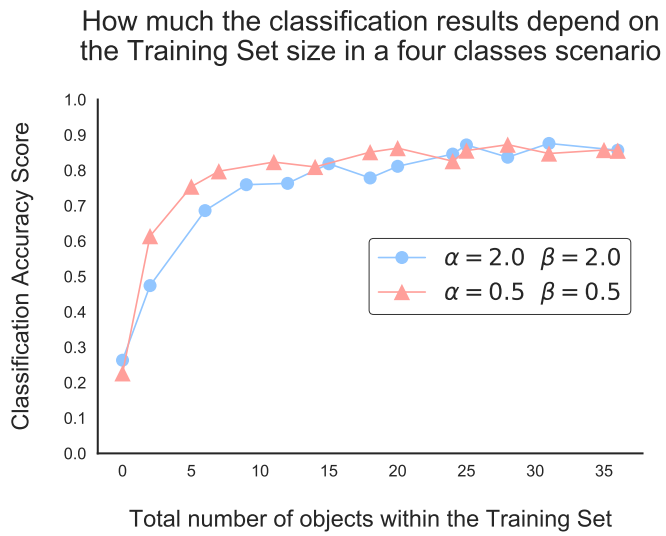


Fig. 8: Classifiers voting for MACHO data. Note that increasing the training set size in about 10 instances produces that the model gets an increment of 50% (from 20% to 80%) in the classification accuracy score. This figure shows that after a small number of instances the accuracy remains more or less stable, independent to the training set size.

7.4 Recovery of Credibility Matrices $\hat{\Theta}$ on MACHO Data

The relation between the accuracy classification score and the training set size are closely related, as we show in figure 8. In addition, figure 9 shows the convergence speed of $\hat{\Theta}$ versus the training set size. From figure 8 we have that the accuracy score depends on the training set size. Figure 9 shows that the convergence of $\hat{\Theta}$ also depends on that size. Therefore, we have that if we can estimate a good prior $\hat{\Theta}$ of Θ, we can converge to a higher classification accuracy score. In consequence, the classification accuracy score depends on the convergence of $\hat{\Theta}$.

7.5 Performance Simulations Depending on Θ Convergence on MACHO Data

Figure 10 shows that the better the model estimates the labelers' credibilities Θ, better is the classification accuracy score.

7.6 Performance Simulations on MACHO Data

As it is shown in figure 11(a), in a four-classes-scenario our method reaches the performance of the ABCD question, when we ask a $\mathrm{Random}(1,4)$ number of YN questions per object to each labeler. The implemented baseline is a Bayesian ABCD

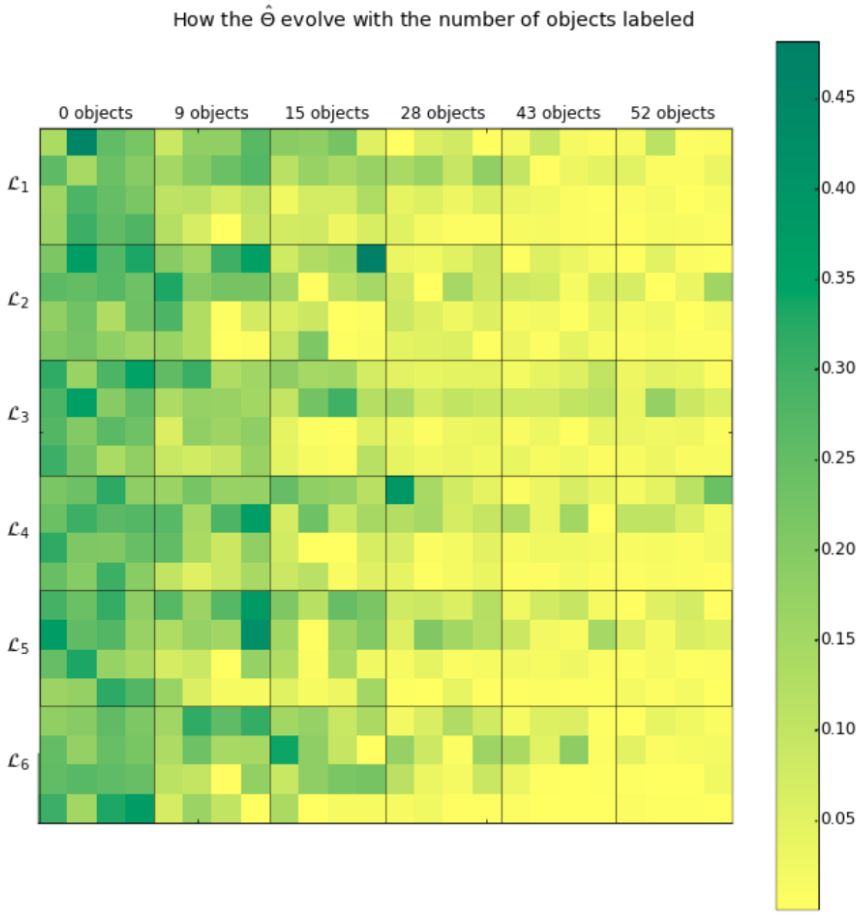


Fig. 9: Classifiers voting for MACHO data MSE. MSE between each original Credibility Matrix row and its recovered $\hat{\theta}^j_{kk'}$ estimation with our method. Note that both figures 8 and 9 converge by the 35 objects. If we have 36 objects and 4 classes, each labeler votes for about 9 objects per class. $\mathrm{E}\{\mathrm{Random}(1,4)\} = 2.5$ questions per object implies 22.5 questions per class, it means about 5.6 votes for estimating each $\hat{\theta}^j_{kk'}$. We can see that the convergence of $\hat{\Theta}$ depends directly on that set size.

model. This baseline is the Hybrid Confusion Matrix (Liu and Wang, 2012) based on the DawidSkene model (Dawid and Skene, 1979) plus a prior estimation stage of confusion matrices.

Figure 11(b) shows the same experiment in a five-classes-scenario. We can see that when all classes are asked per object per labeler, the YN model outperforms the ABCD strategy. However, three labelers are not enough for this scenario because the only way they reached the ABCDE performance (five classes imply ABCDE)

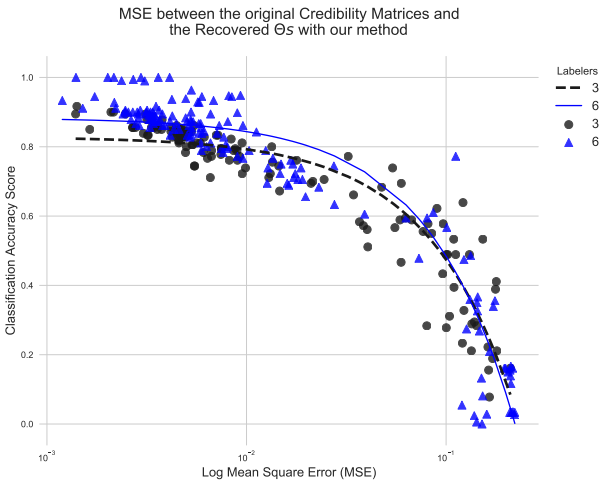


Fig. 10: MSE between original Credibility Matrices and the Recovered ones. Classifiers voting for MACHO data. The error has two possible sources: i) an insufficient size of the training set, and ii) a lack of convergence in the model. In conclusion, how accurate is the estimation of $\mathrm{P}(\mathbf{Z}|\mathbf{R})$ depends on the quality of the estimation of Θ.

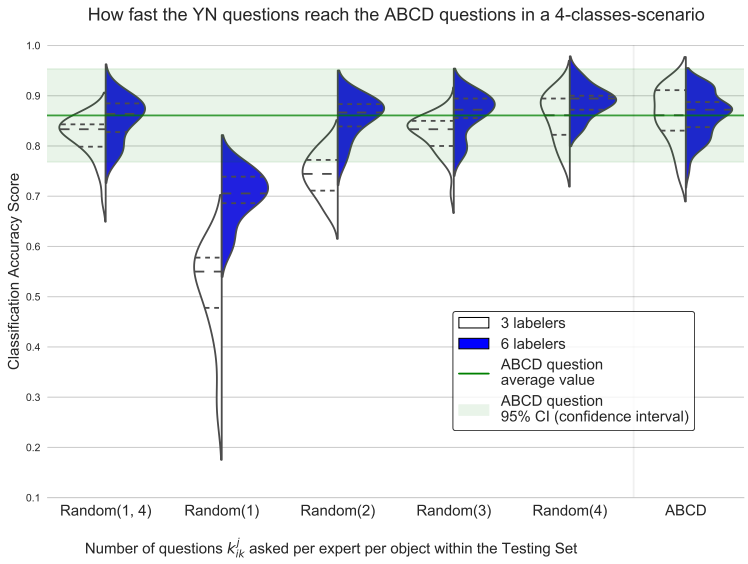
was when we asked them about all the five classes. In this five-classes-scenario, six labelers outperform the ABCDE model when giving responses for only four classes. It means that the labelers were not required to discern among the five classes to reach high accuracy score.

Both scenarios in figure 11 show that the YN model outperforms the ABCD method when we ask a YN question for every possible class $\mathcal{M}_k$ for every object $\mathcal{X}_i$. It gives signals 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. Hence, the YN model can realize with similar confidence intervals the main mistakes made by the labelers.

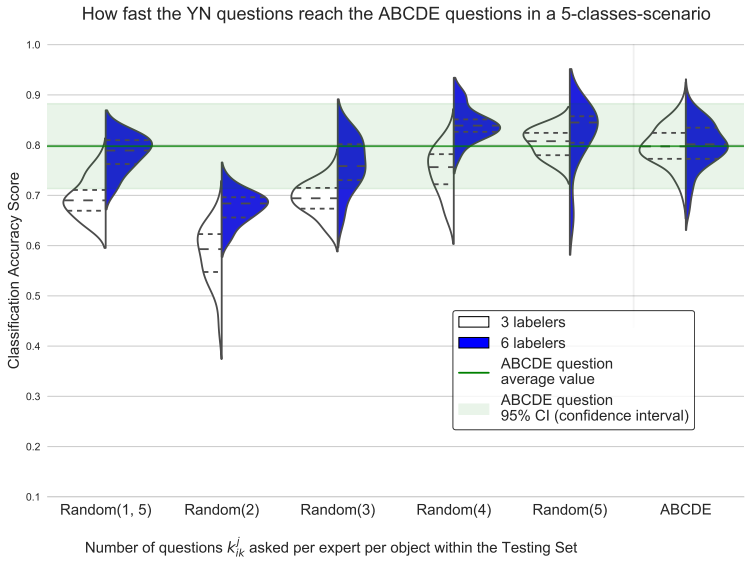
In a six-classes-scenario, three labelers were not enough to achieve the full question performance either. In this case, six labelers reached the ABCDEF question (six classes imply ABCDEF) only when they were asked YN questions for all the possible classes. It means, there was no reduction in the effort under these conditions.

7.7 Performance Real-World Votes MCMC vs. BBVI on Websites Results

We developed all the previous simulations using the PyMC3 implementation mainly for two reasons. First, even though when we used the AdaGrad (Ranganath et al, 2014) algorithm to set the learning rate, this setting presents more parameters tunning



(a) Four-classes-scenario



(b) Five-classes-scenario

Fig. 11: 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.

than the MCMC parametrization. Second, the results where most of the time slightly outperformed by the PyMC3 implementation.

Iterations Until Converge

As we said before, PyMC3 needs about 3000 iterations until converge when running one chain. BBVI needs only 4 iterations, but each iteration implies to estimate the gradient of each latent variable, that means to take samples from the variational approximation distribution of every variable. This estimation converges at 3072 total samples.

Time and Memory Complexity

The model has $\mathcal{J} \times \mathcal{K} \times \mathcal{K} \times 2$ and $\mathcal{J} \times \mathcal{K} \times \mathcal{K} \times 2 + \mathcal{N} \times \mathcal{K} + \mathcal{K}$ parameters to estimate, respectively in each stage. If we assume that always $\mathcal{J} \times \mathcal{K} < \mathcal{N}$, this model is $\Omega(\mathcal{N} \times \mathcal{K})$. For both implementation we need samples. The memory complexity for the PyMC3 model is $\mathcal{O}(\mathcal{N} \times \mathcal{K} \times NumOfSamplesMCMC)$ and for the BBVI is $\mathcal{O}(\mathcal{N} \times \mathcal{K} \times NumOfSamplesBBVI)$, in that case both number of samples are constant, then the complexity is $\mathcal{O}(\mathcal{N} \times \mathcal{K})$. Both time complexities are equivalent.

Time Until Complete Convergence

The experiments performed a time of 10 minutes (PyMC3) versus 5 minutes (BBVI) for The Catalina Survey full model running 1 chain. Moreover, for the Animals Dataset 14 minutes (PyMC3) versus 7 minutes (BBVI) respectively. Taking into account the sizes were equal, and 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.

These statistics are measured on a YN model that uses random sampling to ask a random labeler for labels. Since the YN model uses MCMC sampling, an active learning strategy is impractical in this scenario. Each active learning steps requires converging a model to estimate the next question and labeler. Furthermore, given that the experiments took minutes to converge, the current implementations cannot support an active learning strategy with humans in the loop.

The results for The Catalina Surveys are shown in figure 12. It is possible to observe that for this data the MCMC model outperforms the BBVI implementation. For the Animals Data, both implementations get 99.7% of 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 (see (15) in appendix B for details).

7.8 Performance Crowd Versus Each Labeler on Websites Results

To evaluate the individual performance of each labeler versus the mixture of them, we train one YN model per labeler and via posterior inference evaluate the value of the

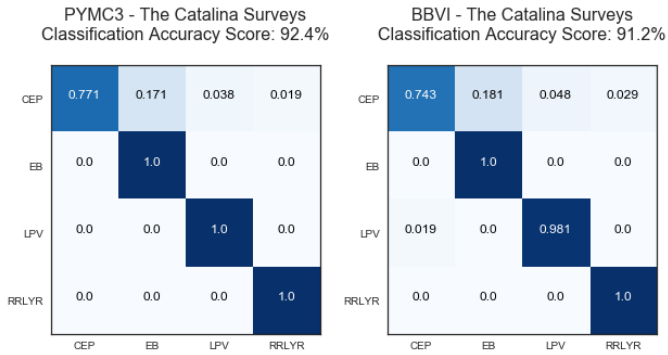


Fig. 12: PyMC3 vs. BBVI. Confusion matrices for the learned models from Real-World Data.

classes as the prediction. Figure 13 shows the individual performance of each labeler selected from the contest with the Catalina Surveys. According to that, our model can effectively model and integrate 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 interpret the labelers responses. Second, the mixture of labelers helps the model to converge to a correct posterior distribution of the classes 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. We do not assume that during the credibility estimation labelers give responses to estimate every $\hat{\theta}^{j}_{kk'}$. This produces that the performance of the YN model with only one labeler can increase if we ask for more training set.

7.9 Performance Real-World Votes YN vs. ABCD on Websites Results

As we explained in section 6, each labeler was presented to a series of full ABCD questions for 80 objects of all data, for which the labelers were asked for $\mathrm{Random}(1,4)$ YN queries as well. Table 2 shows those results.

Table 2 Accuracy scores - Objects with responses YN and ABCD

Contest	*YN query type*	*ABCD query type*
The Catalina Survey	91.2%	90.0%
Animals Dataset	100%	100%

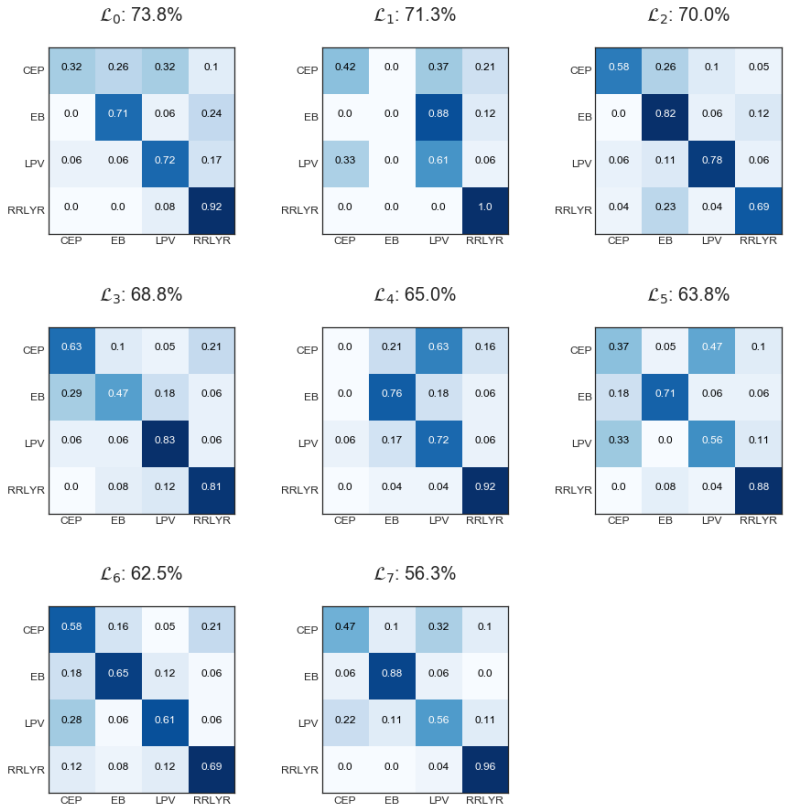


Fig. 13: The Catalina Surveys contest's participants. Even though the labelers are confused, the YN model can learn how the labelers fail. We can observe in figures 12 that our model outperforms each labeler.

7.10 Performance Analysis YN Question vs. ABC Question on Websites Results

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 each ABCD question needs at most two times the time that a YN question needs to be answered. To measure the cost, we compare 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 (Welinder and Perona, 2010), because each ABCD response requires the labeler to recognize the YN response for all the $\mathcal{K}$ possible classes. Figure 14 shows that if 4 YN queries require as much effort as 1 ABCD question does, the YN model converges faster and to a higher classification accuracy score. This occurs because the YN model can differentiate better among the possible

errors since the YN query gives specific information to estimate all the rows within the credibility matrices. And as figure 10 shows, the better the model estimates the credibility matrices, the better the classification accuracy score.

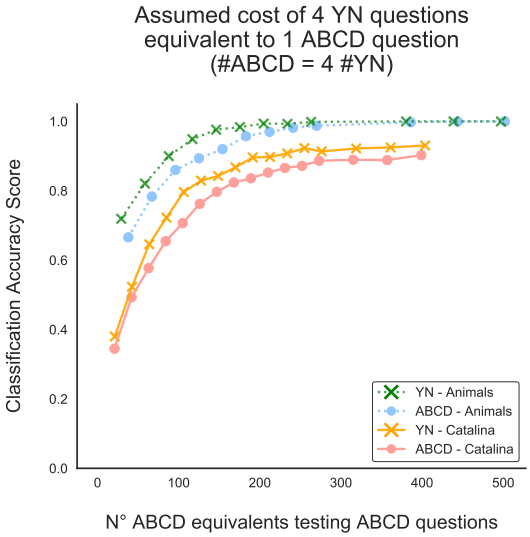


Fig. 14: ABCD equivalent questions. Results from two web contests: Real-World votes on two different scenarios. The ABCD predictions are obtained from the Bayesian model described in section 7.6.

Despite this assumption, that 4 YN queries are equivalent to 1 ABCD query, in figure 15 we present an analysis based on different ABCD equivalents questions. All ABCD predictions are obtained from the Bayesian model described in section 7.6, which is also used in figure 11(a).

The analysis in figure 15 corresponds to how much difference there is 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 14, where the YN method surpasses the ABCD strategy.

We compare this error (axis-Y) to the number of ABCD equivalent questions asked during the labeling stage (axis-X). We can see in figure 15 that the YN strategy outperforms the ABCD strategy when we assume 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 to each labeler the YN model reaches the ABCD performance quickly. Furthermore, when we assume that each YN question is equivalent in cost to one ABCD question, figure 15 shows that at some point the YN question reaches or outperforms the ABCD performance.

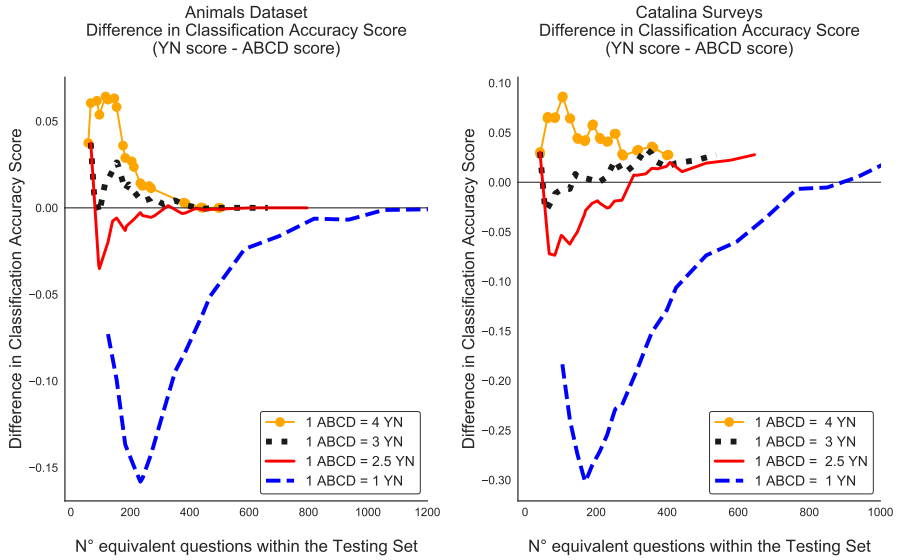


Fig. 15: Difference in Classification Accuracy Score. Results from two web contests: Real-World votes on two different scenarios. The ABCD predictions are obtained from the Bayesian model described in section 7.6.

7.11 Cognitive Cost Analysis YN Question vs. ABC Question on Websites Results

The assumption of cognitive effort made by annotators depends on factors like the information presented to the annotators and the number of classes among others. Since we cannot evaluate all possible scenarios objectively., we show the results of assessing different costs equivalences in a four-classes-scenario in figure 16. It is possible to observe in figure 16 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 a half of 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 YN cognitive cost is less than 0.6 times that of the ABCD, the YN strategy reduces the total effort.

8 Conclusion

We developed a new model for crowdsourcing with “yes” or “no” types of queries. Results show that our model obtains comparable results with models that ask for full questions to labelers. The reduction of the labelers’ effort varies according to the scenario in which the model is applied. It mainly depends on how much cognitively easier is to respond a YN versus an ABCD question in that scenario. From the results, we can see that the effort reduction also comes from the ability of our model to

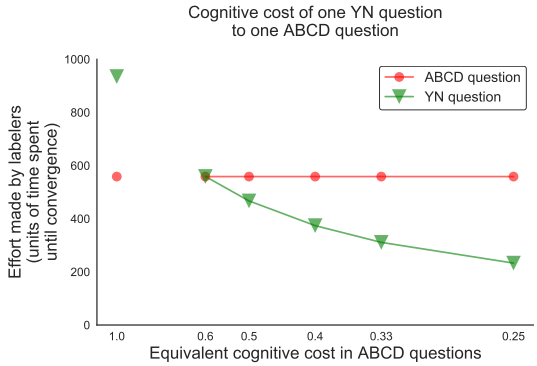


Fig. 16: Results from two web contests. The ABCD predictions are obtained from the Bayesian model described in section 7.6. The y-axis values were taken from the website scenarios. The times marked at cost 1.0 are empirical data, and any other point is proportional to the assumed cognitive effort.

realize quickly the main mistakes made by the labelers. This ability is reflected as a more rapid convergence of the classification accuracy score in a four-classes-scenario (14, 15). Moreover, that fast convergence does not come with a sacrifice in accuracy, because our model can quickly detect most of the classes where the labelers have low confidence. That observation is key to the proposed approach, which shows that detecting where the labelers are weak is as informative as identifying where the labelers are strong.

As it is stated through this work, how much cheaper the YN strategy is for the labelers depends on each scenario settings. Scenarios may vary on the number of objects, classes, and labelers; and in how questions are presented to the labelers. So far, results show that the YN model captures the right labels in a four- and five-classes-scenario with six labelers. However, with six labelers, the YN model accuracy score degrades as the number of classes increases. Hence, as future work, we could estimate the minimum requirements for the YN strategy to work in a given scenario.

Results show that asking for a few known objects the YN model converges to a high classification accuracy score, so once we have enough data to estimate the credibility matrices, we can invest the rest of the time asking the labelers for unknown objects. However, since the sampling times were several minutes for small datasets, we still need to solve how to make this technique scales up for larger datasets.

From the mixture-of-labelers perspective, we can see that in cases where most of the labelers were unreliable in many classes, the full probabilistic model was able to capture the right posterior of the labels by taking advantage of crowds. That occurs mainly because the model learns to map crowds' responses between the credibility estimation stage and the labeling stage. As a future contribution, the YN model should be evaluated in cases where most of the labelers are random voters. Under these conditions, we should use a quality control strategy to filter crowds' responses.

In addition, the model should include the possibility to increase the labelers expertise with time, something that makes much sense in real scenarios. That could be

achieved by introducing an extra variable in the distributions governing the credibility matrices parameters. As iterations increase, the posterior update should take into account the labeler gained expertise. Another interesting idea we could exploit in future work is to consider that the order in which we present the questions may create a bias in the labelers. For example, if we ask if a cereal bar is healthy after asking for lettuce is not the same as asking if a cereal bar is healthy after showing a burger. Objects shown in previous iterations may modify the way that the labelers perceive our questions.

In this work, the YN strategy is evaluated mainly on its ability to gather crowdsourced data. Furthermore, the proposed model can be applied to any context because it models only crowd responses and not objects features; and the inferred labels can be used to train any classifier. Therefore, as future work, we could evaluate the YN model performance depending on the performance of specific classifiers trained with the YN inferred classes.

There is also space for a further contribution in the way we select the labeler and query at each iteration (Wauthier and Jordan, 2011). So far, we randomly select an object along with a class to ask each labeler available for an answer. This election can be optimized by choosing the following labeler according to an expected information gain criterion. In addition, as proposed by Ipeirotis et al (2014), if we choose to train a specific classifier, we could execute active learning on the training of that model to select the next question to present the labelers.

A Background Theory

This section describes the main theory behind this work: how to represent probabilistic joint distributions by using graphs, following by how to approximate distributions with Markov chain Monte Carlo (MCMC) inference and variational inference (VI). We explore deeply VI by presenting the evidence lower bound (ELBO), mean field inference, stochastic variational inference (SVI), and black box variational inference (BBVI). We base our discussion strongly on Blei et al (2017) and Murphy (2012).

A.1 Probabilistic Graphical Models

We represent the joint distribution of the proposed method with a probabilistic graphical model (PGM) (Koller and Friedman, 2009; Wainwright et al, 2008) (see section 3 for our model). A PGM is a graph-based representation for compactly encoding a complex distribution over a high-dimensional space. For example, figure 17 illustrates the elemental DawidSkene (Dawid and Skene, 1979) distribution for a crowdsourcing classification scenario. Where the circles represent random variables, observed variables are gray circles, and the points represent hyperparameters. When a set of variables share the same probability distribution, we can use the "plate" notation, which stacks identical objects in a rectangle representation. In that case, the plates dimensions are written in capital letters within the rectangles.

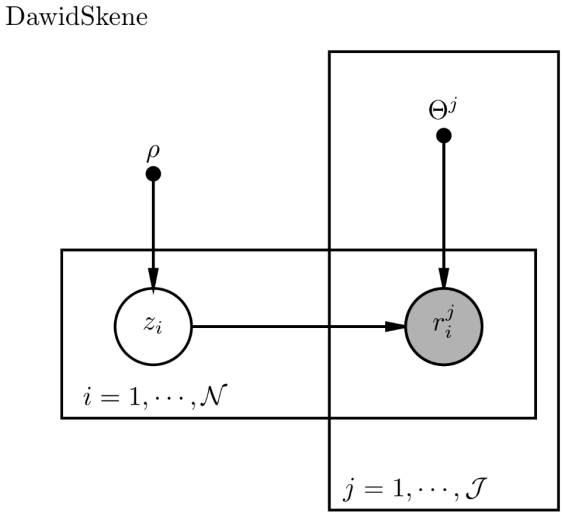


Fig. 17: The DawidSkene model represented as a PGM in plate notation.

In the PGM showed in figure 17, $\mathcal{N}$ is the number of instances to be labeled, and $\mathcal{J}$ is the number of labelers, where $i \in \{1, ..\mathcal{N}\}$ and $j \in \{1, ..\mathcal{J}\}$. In the DawidSkene model, ρ is the initial parameter for the distribution over the hidden labels $\mathbf{Z}$, where z_i is the predicted label for the object $\mathcal{X}_i$. In that scenario, r_i^j represents the class given by labeler L_j to object $\mathcal{X}_i$, whose confusion matrix is Θ^j. In this case, if each Θ^j is a random variable instead of an hyperparameter, $\mathbf{Z}$ and Θ will be conditionally dependent given all the labelers votes $\mathbf{R}$ due to the graph structure. Following the Liu and Wang (2012) notation, in that model definition the variable distributions are:

$$z_i \sim \text{Multinomial}(\rho) \tag{5}$$

$$r_i^j \sim \text{Multinomial}(\Theta^j(z_i, :)) \tag{6}$$

This structure allows us to infer a compact representation of the explicit joint distribution. To get the posterior distribution, we can either use sampling-based methods or variational inference. In this work, we address the proposed probabilistic model (see section 3) solution with approximate inference. In the following subsections, we explain two approaches to infer the posterior target distribution by approximating a distribution: Markov chain Monte Carlo (MCMC) and Variational Inference (VI).

A.2 Markov Chain Monte Carlo

MCMC (Hastings, 1970; Gelfand and Smith, 1990) is the most popular method for sampling when simple Monte Carlo methods do not work well in high-dimensional spaces. The key idea is to build a Markov chain on the state space $\mathcal{Z}$ where the stationary distribution is the target, for instance, a posterior distribution $\mathrm{p}(z|x)$, where x is observed data. MCMC performs a random sampling walk on the $\mathcal{Z}$ space, where the time spent in each state z is proportional to the target distribution. The samples allow us to approximate $\mathrm{p}(z|x)$.

MCMC approaches Bayesian inference with developments as the Gibbs sampler (Geman and Geman, 1984). The key idea behind Gibbs sampling is to turn the sampling among the variables. In each turn the sampler conditions a new variable sample s on the recent values of the rests of the distributions in the model. Suppose we want to infer $\mathrm{p}(z_1, z_2)$. In each iteration, we turn the samples iteratively: $z_1^{s+1} \sim \mathrm{p}(z_1|z_2^s)$ and $z_2^{s+1} \sim \mathrm{p}(z_2|z_1^s)$.

No-U-Turn Hamiltonian Monte Carlo (NUTS)

To avoid the random walk and converge the sampling faster than with simple MCMC, we use NUTS (Hoffman and Gelman, 2014), an MCMC algorithm based on a Hamiltonian Monte Carlo sampler (HMC). As an advantage, NUTS uses an informed walk and avoids the random walk by using a recursive algorithm to obtain a set of candidate points widely spread over the target distribution. Furthermore, NUTS stops when the recursion starts to back and trace the dropped steps again. Nevertheless, HMC requires computing the gradient of the log-posterior to inform the walk, which can be hard.

Using NUTS does not oblige to establish the step size and the number of steps to converge, compared to what a simple MCMC or HMC sampler does. Setting those parameters would require preliminary runs and some expertise. This sampling stops when drawing more samples does no longer increase the distance between the proposal $\tilde{z}$ and the initial values of z.

Even though MCMC algorithms can be very slow when working with large datasets or very complex models, they asymptotically draw exact samples from the target density (Robert, 2004). Under these heavy computational settings, we can use variational inference (VI) as an approximation to the target distribution. VI does not guarantee to find the density distribution, it only finds a close distribution, but usually, it is faster than MCMC.

A.3 Variational Inference

Variational inference (VI) (Jordan et al, 1999; Wainwright et al, 2008) proposes a solution to the problem of posterior inference. VI selects an approximation $\mathrm{q}(z)$ from some tractable family and then it tries to make this $\mathrm{q}(z)$ as close as possible to the true posterior $\mathrm{p}^*(z) \triangleq \mathrm{p}(z|x)$. The VI approach reduces this approximation to an optimization problem, the minimization of the KL divergence (Kullback and Leibler, 1951) from q to p^*.

The KL divergence is a measure of dissimilarity of two probability distributions, p^* and q. Given that the forward KL divergence $\mathbb{KL}(\mathrm{p}^*||\mathrm{q})$ includes taking expectation over the intractable $\mathrm{p}^*(z)$, a natural alternative is the reverse KL divergence $\mathbb{KL}(\mathrm{q}||\mathrm{p}^*)$, defined in (7).

$$\mathbb{KL}(\mathrm{q}||\mathrm{p}^*) = -\int \mathrm{q}(z) \log \frac{\mathrm{q}(z)}{\mathrm{p}^*(z)} dz \tag{7}$$

A.4 The Evidence Lower Bound

Variational inference minimizes the KL divergence from q to p^*. It can be shown to be equivalent to maximize the lower bound (ELBO) on the log-evidence $\log \mathrm{p}(x)$. The ELBO is equivalent to the negative KL divergence plus a constant, as we show in the following definitions.

Lets say x are the observations, z the latent variables, and λ the free parameters of $\mathrm{q}(z|\lambda)$. We want to approximate $\mathrm{p}(z|x)$ by setting λ such as the KL divergence is minimum. In this case we can rewrite (7), and expand the conditional in (8).

$$\mathbb{KL}(\mathrm{q}||\mathrm{p}^*) = \mathbb{E}_{\mathrm{q}}[\log \mathrm{q}(z|\lambda)] - \mathbb{E}_{\mathrm{q}}[\log \mathrm{p}(z,x)] - \log \mathrm{p}(x) \tag{8}$$

Therefore, the minimization of the KL in (9) is equivalent to maximizing the ELBO:

$$\mathcal{L}(\mathrm{q}) = \mathbb{E}_{\mathrm{q}}[\log \mathrm{p}(z,x) - \log \mathrm{q}(z|\lambda)] \tag{9}$$

A.5 Mean Field Inference

The optimization over a given family of distributions is determined by the complexity of the family. This optimization can be as difficult to optimize as complex is the family used. To keep the variational inference approach simple, Opper and Saad (2001) proposes to use the mean field approximation. This approach assumes that the posterior can be approximated by a fully factorized q, where each factor is an independent mean field variational distribution, as it is defined in (10).

$$\mathrm{q}(z) = \prod_{i=1}^{m} \mathrm{q}_i(z_i) \tag{10}$$

The goals is to solve the optimization in (11) over the parameters of each marginal distribution q.

$$\min_{\lambda_1,\ldots,\lambda_m} \mathbb{KL}(\mathrm{q}||\mathrm{p}^*) \tag{11}$$

A.6 Stochastic Variational Inference

Common posterior inference algorithms do not easily scale to work with high amounts of data. Furthermore, several algorithms are very computationally expensive because they require passing through the full dataset in each iteration. Under these settings, stochastic variational inference (SVI) (Hoffman et al, 2013) approximates the posterior distribution by computing and following its gradient in each iteration over subsamples of data. SVI iteratively takes samples from the full data, computes its optimal local parameters, and finally, it updates the global parameters.

SVI solves the ELBO optimization by using the natural gradient (Amari, 1998) in a stochastic optimization algorithm. This optimization consists in to estimate a noisy but cheap to compute gradient to reach the target distribution.

A.7 Black Box Variational Inference

The BBVI (Ranganath et al, 2014) avoids any model-specific derivations. Black Box VI proposes to stochastically maximize the ELBO using noisy estimates of its gradient. The estimator of this gradient is computed using samples from the variational posterior. Then, we need to write the gradient of the ELBO (9) in (12).

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\mathrm{q}}[\nabla_\lambda \log \mathrm{q}(z|\lambda)(\log \mathrm{p}(z,x) - \log \mathrm{q}(z|\lambda))] \tag{12}$$

Using this equation, we can compute the noisy unbiased gradient of the ELBO sampling the variational distribution with Monte Carlo, as it is showed in equation (13), where S is the number of samples we take from each distribution to be estimated.

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \log \mathrm{q}(z_s|\lambda)(\log \mathrm{p}(z_s, x) - \log \mathrm{q}(z_s|\lambda)) \tag{13}$$

where,

$$z_s \sim \mathrm{q}(z|\lambda) \tag{14}$$

For estimating the approximating q distribution, in BBVI the variational distributions $\mathrm{q}(z_i)$ are mean field factors with free variational parameters λ_i, for each index i (see (10)). In appendix B we show how to apply this method to the proposed model.

B Derivation Black Box Inference Equations

The BBVI (Ranganath et al, 2014) minimizes the KL divergence from an approximating distribution q to the true p posterior. Lets say x are the observations, z latent variables, and λ the free parameters of $\mathrm{q}(z|\lambda)$. And we want to approximate $\mathrm{p}(z|x)$ by setting λ. This optimization is equivalent to maximizing the ELBO in (9):

$$\mathcal{L}(\lambda) = \mathbb{E}_\mathrm{q}[\log \mathrm{p}(z, x) - \log \mathrm{q}(z|\lambda)]$$

BBVI proposes stochastically maximize the ELBO using noisy estimates of its gradient. The estimator of this gradient is computed using samples from the variational posterior. Then, we need to write the gradient of the ELBO in (12):

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_\mathrm{q}[\nabla_\lambda \log \mathrm{q}(z|\lambda)(\log \mathrm{p}(z, x) - \log \mathrm{q}(z|\lambda))]$$

Using (12), we can compute the noisy unbiased gradient of the ELBO sampling the variational distribution with Monte Carlo, as it is showed in (13), where S is the number of samples we take from each distribution to be estimated:

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \log \mathrm{q}(z_s|\lambda)(\log \mathrm{p}(z_s, x) - \log \mathrm{q}(z_s|\lambda))$$

$$\text{Where, } z_s \sim \mathrm{q}(z|\lambda)$$

Then λ is set at each iteration t as:

$$\lambda_t = \lambda_{t-1} + \rho \nabla_\lambda \mathcal{L} \tag{15}$$

Where, the learning rate ρ can be fine-tuned as a global rate for all λs or as a unique rate per λ_s.

To estimate the approximating q distribution, BBVI uses the mean field theory. Then we define the approximating distribution q as in (10):

$$\mathrm{q}(z) = \prod_{i}^{m} \mathrm{q}_i(z_i)$$

The variational mean field distributions q from (10) in the Credibility Estimation (first stage) of the YN model are in (16). Whose free variational parameters to estimate are in (17).

$$\mathrm{q}(\hat{\theta}^j_{kk'}) \sim \mathrm{Beta}(\hat{\alpha}^j_{kk'}, \hat{\beta}^j_{kk'}) \; \forall \; jkk' \tag{16}$$

$$\hat{\Theta} : \; (\hat{\alpha}^j_{kk'}, \hat{\beta}^j_{kk'}) \; \forall \; \hat{\theta}^j_{kk'} \tag{17}$$

For the Labeling part (second stage) of the proposed model, the mean field distributions q from (10) are defined in (18), (19) and (20).

$$\mathrm{q}(\theta^j_{kk'}) \sim \mathrm{Beta}(\alpha^j_{kk'}, \beta^j_{kk'}) \; \forall \; jkk' \tag{18}$$

$$\mathrm{q}(z_i) \sim \mathrm{Categorical}(\mathbf{p}_i) \,\forall\, i \tag{19}$$

$$\mathrm{q}(\pi) \sim \mathrm{Dirichlet}(\mathbf{d}) \tag{20}$$

Whose free variational parameters to estimate are in (21), (22), and (23) respectively.

$$\Theta:\ (\alpha^j_{kk'}, \beta^j_{kk'}) \,\forall\, \theta^j_{kk'} \tag{21}$$

$$\mathbf{z}:\ (p_{ik}\ \forall k \in K) \,\forall\, z_i \tag{22}$$

$$\pi:\ (d_k\ \forall k \in K) \tag{23}$$

As it is shown in (12), to maximize the ELBO, we need the expectations under q. Given that we prefer to avoid that derivation for the YN model joint distribution, we use the black box method by approximating the gradient of the ELBO as defined in (13).

For applying this method to our model, we need to write the needed functions for the Credibility stage and the Labeling stage as well. In this appendix, we show only the derivation for that second stage (the gradients for the training part are a simplification of the presented derivations).

B.1 Labeling Parameters Estimation

The joint distribution to be inferred is:

$$\log \mathrm{p}(r^j_{ik}, z_i, \theta^j_{z_i k}, \pi) = \\ \log \mathrm{p}(\theta^j_{z_i k}|\alpha_0, \beta_0) + \sum_{i=1}^{\mathcal{N}} \left\{ \log \mathrm{p}(z_i|\theta^j_{z_i k}, \pi) + \log \mathrm{p}(r^j_{ik}|z_i, \theta^j_{z_i k}) \right\} \tag{24}$$

First, for each variable, we define the log probability of all distributions containing the free parameters in order to obtain the mean field q. The priors are:

$$\log \mathrm{p}(\theta^j_{kk'}|\alpha_0, \beta_0) = \log \mathrm{Beta}(\theta^j_{kk'}|\alpha_0, \beta_0) \tag{25}$$

$$\log \mathrm{p}(z_i|\theta^j_{kk'}, \pi) = \log \mathrm{Categorical}(z_i|\pi) \tag{26}$$

$$\log \mathrm{p}(\pi|\rho) = \log \mathrm{Dirichlet}(\pi|\rho) \tag{27}$$

$$\log \mathrm{p}(r^j_{ik}|z_i, \theta^j_{z_i k}) = \log \left\{ (\theta^j_{z_i k})^{r^j_{ik}[0]} \times (1 - \theta^j_{z_i k})^{r^j_{ik}[1]} \right\} \tag{28}$$

Then, we can write those log probabilities to estimate the gradient with respect to the variational parameters:

$$\log \mathrm{q}(\theta^j_{kk'}|\alpha^j_{kk'}, \beta^j_{kk'}) = \log \mathrm{Beta}(\theta^j_{kk'}|\alpha^j_{kk'}, \beta^j_{kk'}) = \\ \frac{\Gamma(\alpha^j_{kk'} + \beta^j_{kk'})}{\Gamma(\alpha^j_{kk'})\Gamma(\beta^j_{kk'})} \times (\theta^j_{kk'})^{\alpha^j_{kk'}-1} \times (1 - \theta^j_{kk'})^{\beta^j_{kk'}-1} \tag{29}$$

$$\log \mathrm{q}(z_i|\mathbf{p}_i) = \log \mathrm{Categorical}(z_i|\mathbf{p}_i) = \sum_{k=1}^{\mathcal{K}} \{z_i k \times \log p_{ik}\} \tag{30}$$

$$\log \mathrm{q}(\pi|\mathbf{d}) = \log \mathrm{Dirichlet}(\pi|\mathbf{d}) = \sum_{k=1}^{\mathcal{K}} \{(d_k - 1) \times \log \pi_k\} \tag{31}$$

Finally, we can write the gradients for each parameter to be estimated, where $\Psi(x) = \frac{d\Gamma(x)}{dx}$:

$$\nabla_{\alpha^j_{kk'}} \log \mathrm{q}(\theta^j_{kk'}|\alpha^j_{kk'}, \beta^j_{kk'}) = \log \theta^j_{kk'} + \Psi(\alpha^j_{kk'} + \beta^j_{kk'}) - \Psi(\alpha^j_{kk'}) \tag{32}$$

$$\nabla_{\beta^j_{kk'}} \log \mathrm{q}(\theta^j_{kk'}|\alpha^j_{kk'}, \beta^j_{kk'}) = \log (1 - \theta^j_{kk'}) + \Psi(\alpha^j_{kk'} + \beta^j_{kk'}) - \Psi(\beta^j_{kk'}) \tag{33}$$

$$\nabla_{p_{ik}} \log \mathrm{q}(z_i|\mathbf{p}_i) = \frac{z_{ik}}{p_{ik}} \tag{34}$$

$$\nabla_{d_k} \log \mathrm{q}(\pi|\mathbf{d}) = \log d_k - \Psi(d_k) - \Psi(\sum_{k=1}^{\mathcal{K}} d_k) \tag{35}$$

B.2 Constrain Parameters

All the estimated parameters must be positive to remain in their distribution domain. In fact, each vector $\mathbf{p}_i$ and the vector $\mathbf{d}$ must sum one. We use the soft-plus function, and a normalized soft-plus function to deal with these constrains.

Acknowledgements Work supported in part by the CSS survey. The CSS survey is funded by the National Aeronautics and Space Administration under Grant No. NNG05GF22G issued through the Science Mission Directorate Near-Earth Objects Observations Program. The CRTS survey is supported by the U.S. National Science Foundation under grants AST-0909182 and AST-1313422.

References

Amari SI (1998) Natural gradient works efficiently in learning. Neural computation 10(2):251–276

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. Journal of machine Learning research 3(Jan):993–1022

Blei DM, Kucukelbir A, McAuliffe JD (2017) Variational inference: A review for statisticians. Journal of the American Statistical Association (just-accepted)

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

Bragg J, Weld DS, et al (2016) Optimal testing for crowd workers. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems, pp 966–974

Carpenter B (2011) A hierarchical bayesian model of crowdsourced relevance coding. In: TREC

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

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

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

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

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

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

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

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

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

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

Hoffman MD, Blei DM, Wang C, Paisley J (2013) Stochastic variational inference. The Journal of Machine Learning Research 14(1):1303–1347

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

Ipeirotis PG, Provost F, Sheng VS, Wang J (2014) Repeated labeling using multiple noisy labelers. Data Mining and Knowledge Discovery 28(2):402–441

Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999) An introduction to variational methods for graphical models. Machine learning 37(2):183–233

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

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

Kullback S, Leibler RA (1951) On information and sufficiency. The annals of mathematical statistics 22(1):79–86

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

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

Liu Q, Peng J, Ihler AT (2012) Variational inference for crowdsourcing. In: Advances in neural information processing systems, pp 692–700

Moreno PG, Artés-Rodríguez A, Teh YW, Perez-Cruz F (2015) Bayesian nonparametric crowdsourcing. Journal of Machine Learning Research

Murphy KP (2012) Machine learning: a probabilistic perspective. MIT press

Opper M, Saad D (2001) Advanced mean field methods: Theory and practice. MIT press

Otani N, Baba Y, Kashima H (2015) Quality control for crowdsourced hierarchical classification. In: Data Mining (ICDM), 2015 IEEE International Conference on, IEEE, pp 937–942

Pichara K, Protopapas P (2013) Automatic classification of variable stars in catalogs with missing data. The Astrophysical Journal 777:83, 10.1088/0004-637X/777/2/83, URL http://m.iopscience.iop.org/0004-637X/777/2/83/metrics

Pichara K, Soto A (2011) Active learning and subspace clustering for anomaly detection. Intelligent Data Analisys 15(2):151–171

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

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

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

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

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, URL http://papers.nips.cc/paper/6523-data-programming-creating-large-training-sets-quickly.pdf

Raykar VC, Yu S, Zhao LH, Jerebko A, Florin C, Valadez GH, Bogoni L, Moy L (2009) Supervised learning from multiple experts: whom to trust when everyone lies a bit. In: Proceedings of the 26th Annual international conference on machine learning, ACM, pp 889–896

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

Robert CP (2004) Monte carlo methods. Wiley Online Library

van Rossum G, (eds) FD (2001) Python reference manual. URL http://www.python.org

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al (2014) Imagenet large scale visual recognition challenge. arXiv preprint arXiv:14090575

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

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

Settles B (2010) Active learning literature survey. University of Wisconsin, Madison 52(55-66):11

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

Vondrick C, Patterson D, Ramanan D (2013) Efficiently scaling up crowdsourced video annotation. International Journal of Computer Vision 101(1):184–204

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

Wauthier FL, Jordan MI (2011) Bayesian bias mitigation for crowdsourcing. In: Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira F, Weinberger KQ (eds) Advances in Neural Information Processing Systems 24, Curran Associates, Inc., pp 1800–1808, URL http://papers.nips.cc/paper/4311-bayesian-bias-mitigation-for-crowdsourcing.pdf

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
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
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
Yan Y, Rosales R, Fung G, Dy J (2012) Modeling multiple annotator expertise in the semi-supervised learning scenario. arXiv preprint arXiv:12033529
Zhang C, Chaudhuri K (2015) Active learning from weak and strong labelers. In: Advances in Neural Information Processing Systems, pp 703–711
Zhang Y, Chen X, Zhou D, Jordan MI (2014) Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In: Advances in neural information processing systems, pp 1260–1268