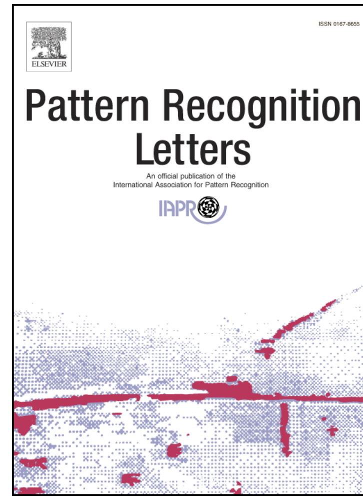

Learning from Multiple Annotators: Distinguishing Good from Random Labelers

Filipe Rodrigues, Francisco Pereira, Bernardete Ribeiro

PII: S0167-8655(13)00202-X
DOI: http://dx.doi.org/10.1016/j.patrec.2013.05.012
Reference: PATREC 5729

To appear in: *Pattern Recognition Letters*

Please cite this article as: Rodrigues, F., Pereira, F., Ribeiro, B., Learning from Multiple Annotators: Distinguishing Good from Random Labelers, *Pattern Recognition Letters* (2013), doi: http://dx.doi.org/10.1016/j.patrec.2013.05.012

Learning from Multiple Annotators: Distinguishing Good from Random Labelers

Filipe Rodrigues[1], Francisco Pereira[2] and Bernardete Ribeiro[1]

[1] *Centre for Informatics and Systems of the University of Coimbra (CISUC)*
Department of Informatics Engineering, University of Coimbra
3030-290 Coimbra, Portugal
Tel.: +351 239790056
{fmpr,bribeiro}@dei.uc.pt

[2] *Singapore-MIT Alliance for Research and Technology (SMART)*
1 CREATE Way, Singapore 138602
Tel.: +65 93233653
camara@smart.mit.edu

Abstract

With the increasing popularity of online crowdsourcing platforms such as Amazon Mechanical Turk (AMT), building supervised learning models for datasets with multiple annotators is receiving an increasing attention from researchers. These platforms provide an inexpensive and accessible resource that can be used to obtain labeled data, and in many situations the quality of the labels competes directly with those of experts. For such reasons, much attention has recently been given to annotator-aware models. In this paper, we propose a new probabilistic model for supervised learning with multiple annotators where the reliability of the different annotators is treated as a latent variable. We empirically show that this model is able to achieve state of the art performance, while reducing the number of model parameters, thus avoiding a potential overfitting. Furthermore, the proposed model is easier to implement and extend to other classes of learning problems such as sequence

labeling tasks.

Keywords: Multiple Annotators, Crowdsourcing, Latent Variable Models, Expectation-Maximization, Logistic Regression

1. Introduction

Crowdsourcing (Howe, 2008) is rapidly changing the way datasets are built. With the development of crowdsourcing platforms such as Amazon Mechanical Turk (AMT)[1], it is becoming increasingly easier to obtain labeled data for a wide range of tasks covering different areas such as Computer Vision, Natural Language Processing, Speech Recognition, etc. The attractiveness of these platforms comes not only from their low cost and accessibility, but also from the surprisingly good quality of the labels obtained, which in many cases competes directly with those of "experts" (Snow et al., 2008). Furthermore, by distributing the workload among multiple annotators, labeling tasks can be completed much faster.

The current trend of social web, where citizens' participation is growing in many forms, has come to stay, and information is being produced at a massive rate. This information can take many forms: document tags, opinions, product ratings, user clicks, contents, etc. These new sources of data also motivate the development of new machine learning approaches for learning from multiple sources.

On another perspective, there are tasks for which ground truth labels simply cannot be obtained due to their highly subjective nature. Consider

[1] http://www.mturk.com

for instance the tasks of sentiment analysis, movie rating or keyphrase extraction. These tasks are subjective in nature and hence no absolute gold standard can be defined. In such cases the only attainable goal is to build a model that captures the *wisdom of the crowds* (Surowiecki, 2004) as well as possible. For such tasks crowdsourcing platforms like AMT become a natural solution. However, the large amount of labeled data needed to compensate for the heterogeneity of annotators' expertise can rapidly rise its actual cost beyond acceptable values. Since different annotators have different levels of expertise, it is important to consider how *reliable* the annotators are when learning from their answers, and a parsimonious solution needs to be designed that is able to deal with such real world constraints (e.g. annotation cost) and heterogeneity.

Even in situations where a ground truth can be obtained, it may be too costly. For example, in Medical Diagnosis, determining whether a patient has cancer may require a biopsy, which is an invasive procedure, and thus should only be used as a last resource. On the other hand, it is rather easy for a diagnostician to consult its colleagues for their opinions before making a decision. Therefore, although there is no crowdsourcing involved here, there are still multiple experts, with different levels of expertise, providing their own (possibly wrong) opinions, from which we have to be able to learn from.

Many approaches have recently been proposed that deal with this increasingly important problem of supervised learning from multiple annotators in different paradigms: classification (Raykar et al., 2009; Yan et al., 2011), regression (Groot et al., 2011), ranking (Wu et al., 2011), etc. However, most of the work developed so far is centered on the *unknown* true labels of the data, for which noisy versions are provided by the various annotators. Therefore, there has been a tendency to include these *unobserved* true labels as latent variables in a probabilistic framework, which, as we demonstrate, is not necessarily the best option. Furthermore, this choice of latent variables hinders a natural extension of these approaches to structured prediction problems such as sequence labeling tasks due to combinatorial explosion of possible outcomes of the latent variables. Contrarily to these approaches, we argue that the focus should be on the annotators, and that including the also *unknown* reliabilities of the annotators as latent variables can be preferable, since it not only leads to simpler models that are less prone to overfitting, but also bypasses the problem of the high number of possible labelings to marginalize over.

In this paper, we propose a new probabilistic model that explores these ideas, and explicitly handles the annotators' reliabilities as latent variables. We empirically show, using both simulated annotators and human annotators from AMT, that for many tasks the new model can be competitive with the state of the art methods, and can even significantly outperform previous approaches under certain conditions. Although we focus on multi-class Logistic Regression as the base classifier, the proposed model is simple and generic enough to be implemented with other classifiers. Furthermore the extension to structured prediction problems such as sequence labeling tasks can be much easier than with latent ground truth models (e.g. Raykar et al. (2010); Yan et al. (2011)).

The remainder of this paper is organized as follows: Section 2 provides the reader with an overview of state of the art; Section 3 clarifies the problem with latent ground truth models; Section 4 presents the proposed model, and Section 5 compares the results obtained by this model with two majority voting baselines and a state of the art approach; the article will end with a short discussion and conclusions (Section 6).

2. State of the art

There is considerable work on estimating ground truth labels from the responses of multiple annotators. Most of the early important works were in the fields of Biostatistics and Epidemiology. In 1979, Dawid and Skene (1979) proposed an approach for estimating the error rates of multiple patients (annotators) given their responses (labels) to multiple medical questions. However, like most of the early works with multiple annotators, this work only focused on estimating the unobserved ground truth labels. Only later, researchers started paying more attention to the specific problem of learning a classifier from the multiple annotator's data. In 1995, Smyth et al. (1995) proposed a similar approach to the one from Dawid and Skene (1979) to estimate the ground truth from the labels of multiple experts, which was then used to train a classifier. As with previous works, the authors employed a model where the unknown true labels were treated as latent variables.

More recently, with the increasing popularity of AMT and other crowd-sourcing and work-recruiting platforms, researchers started recognizing the importance of the problem of learning from the labels of multiple non-expert annotators. The researchers' interest grew even further with works such as (Snow et al., 2008) and (Novotney and Callison-Burch, 2010), which show that, for many tasks, learning from multiple non-experts can be as good as learning from an expert.

With the rising interest in crowdsourcing as a source of labeled data, more challenging approaches for learning from multiple annotators started to appear. In 2009, Raykar et al. (2009) proposed an innovative probabilistic approach where the unknown ground truth labels and the classifier are learnt jointly. By handling the unobserved ground truth labels as latent variables, the authors are able to find the maximum likelihood parameters for their model by iteratively estimating the posterior distribution of the ground truth labels and then using this estimate to determine the qualities of the annotators and the parameters of a Logistic Regression model. Unlike most of the previous works, this approach also has the advantage of relaxing the requirement of repeated labeling, i.e. the same instance being annotated by multiple annotators. Later works then relaxed other assumptions made by the authors. For example, Yan et al. (2010) relaxed the assumption that the quality of the labels provided by the annotators does not depend on the instance they are labeling.

This main line of work also inspired many variations and extensions in the past couple of years. Groot et al. (2011) proposed an extension of Gaussian processes to do regression in a multiple annotator setting. In the field of ranking, Wu et al. (2011) presented an approach to learn how to rank from the opinions of multiple annotators. In an active learning setting, Yan et al. (2011) proposed an approach for multiple annotators by providing answers to the following questions: what instance should be selected to be labeled next and which annotators should label it? On a different perspective, in (Donmez et al., 2010) the authors propose the use of a particle filter to model the time-varying accuracies of the different annotators. Despite the plausibility of their assumptions, i.e. it is legitimate to assume that the quality of the labels provided by an annotator will vary with time, the results obtained showed only a small improvement on the performance of their model through the inclusion of this time dependance.

The approaches above mentioned typically treat the *unknown* ground truth labels as latent variables and build a model on that basis. We argue that explicitly handling the reliabilities of the annotators as latent variables, as opposed to the true labels, in a fashion that slightly resembles a mixture of experts (Jacobs et al., 1991; Bishop, 2006), brings many attractive advantages and can, under certain conditions, outperform latent ground truth models.

3. The problem with latent ground truth models

In order to help motivate the proposed model, we now introduce a typical class of approaches for learning from multiple annotators, in which the *unknown* true labels are treated as latent variables (e.g. Raykar et al. (2009, 2010); Yan et al. (2010)).

Let y_i^r be the label assigned to instance $\mathbf{x}_i$ by the r^{th} annotator, and let y_i be the true (unobserved) label for that instance. Contrarily to a typical classification problem with a single annotator, in a setting with R annotators, a dataset $\mathcal{D}$ with size N consists of a set of labels $\{y_i^1, y_i^2, ..., y_i^R\}$ for each of the N instances $\mathbf{x}_i$.

In general, the class of models we refer to as "latent ground truth models" tend to assume the following generative process: for each instance $\mathbf{x}_i$ there is an *unobserved* true label y_i, and each of the different annotators in-

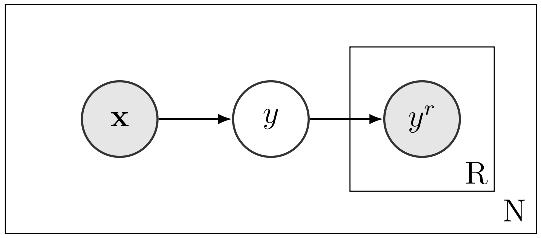

Figure 1: Plate representation of general latent ground truth model.

dependently provides its own version (y_i^r) of this true label, which in practice corresponds to an approximation to the real label y_i. Figure 1 depicts such a model in plate notation. Shaded nodes represent observed variables, and non-shaded nodes represent unobserved (latent) variables.

If besides the dataset $\mathcal{D} = \{y_i^1, ..., y_i^R, \mathbf{x}_i\}_{i=1}^N$ we were given the true labels $\mathcal{Y} = \{y_i\}_{i=1}^N$ as well, the likelihood for this model would take the form

$$p(\mathcal{D}, \mathcal{Y}) = \prod_{i=1}^{N} \Big(p(y_i|\mathbf{x}_i) \prod_{r=1}^{R} p(y_i^r|y_i) \Big). \tag{1}$$

Since we do not actually observe the true labels y_i we must treat them as latent variables and marginalize them out of the likelihood, and this leads us to the first problem with this approach: although this marginalization is not difficult for classification problems where the number of classes (K) is small, for other types of problems like sequence labeling tasks (or any task with structured outputs), marginalizing over the output space is intractable in general (Sutton, 2012). If we consider, for example, the tasks of part-of-speech (POS) tagging or Named Entity Recognition (NER), which are usually handled as a sequence labelling problems, it is easy to see that the number of possible label sequences grows exponentially with the length of the sentence, deeming the marginalization over the output space intractable.

The second problem with this class of models is related with the probability $p(y_i^r|y_i)$, which for a classification problem with K classes requires a $K \times K$ table of parameters for each annotator. Even though this approach allows to capture certain biases in the annotators answers, like for example the tendency to confuse two classes, in practice, on a crowdsourcing platform like AMT, each annotator only labels a rather small set of instances. Therefore, under such conditions, having a model with so many parameters for the reliability of the annotators can easily lead to overfitting. Consider, for example, a classification problem with 10 classes. Such a problem requires a total of 100 parameters (a 10×10 probability table) to model the expertise of a single annotator. To effectively learn such a number of parameters, each annotator would be required to label a large number of instances, at least in the order of the thousands, something that is both unrealistic and hard to control in a crowdsourcing platform.

Taking these issues into consideration, we developed a new probabilistic model for learning from multiple annotators, which we present in the following section.

4. Proposed model

4.1. Maximum likelihood estimator

Given a dataset $\mathcal{D} = \{y_i^1, ..., y_i^R, \mathbf{x}_i\}_{i=1}^N$ with N instances and R different annotators, and assuming that the instances are independent and identically distributed (i.i.d.), the likelihood is given by

$$p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(y_i^1, ..., y_i^R|\mathbf{x}_i, \theta) \tag{2}$$

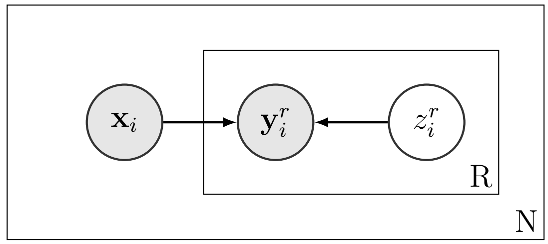

Figure 2: Plate representation of the proposed model.

where θ denotes the model parameters.

Let us now assume the following generative process of the annotators' labels: when the annotators are asked to provide a label to a given instance $\mathbf{x}_i$, they flip a biased coin, and based on the outcome of those coin flips, they decide whether or not to provide the correct label. This intuition amounts to introducing a binary random variable z_i^r, whose value indicates whether the r^{th} annotator labeled the i^{th} instance correctly or not. Hence, $z_i^r \sim Bernoulli(\pi_r)$, where π_r is the accuracy of the r^{th} annotator, and

$$p(z_i^r|\pi_r) = (\pi_r)^{z_i^r}(1-\pi_r)^{1-z_i^r}. \tag{3}$$

The expectation of this Bernoulli random variable $\mathbb{E}[z_i^r] = p(z_i^r = 1)$ can be interpreted as the probability of an annotator providing a correct label or, in other words, as an indicator of how reliable an annotator is. For the sake of simplicity, we assume that an unreliable annotator provides labels according to some random model $p_{\text{Rand}}(y_i^r = k|\mathbf{x}_i)$.

Figure 2 shows a plate representation of this generative model. Notice that the variables z_i^r are not observed in this model, hence their nodes are not shaded in the figure.

If we were told the true values for $\mathcal{Z} = \{z_i^1, ..., z_i^R\}_{i=1}^N$, and assuming the annotators make their decisions independently of the each other, the complete-data likelihood could then be factored as

$$p(\mathcal{D}, \mathcal{Z}|\theta) = \prod_{i=1}^{N} \prod_{r=1}^{R} p(z_i^r|\pi_r)\ p(y_i^r|\mathbf{x}_i, z_i^r, \mathbf{w}) \quad (4)$$

where $\theta = \{\boldsymbol{\pi}, \mathbf{w}\}$ are the model parameters. The values of $\boldsymbol{\pi} = \{\pi_r\}_{r=1}^{R}$ correspond to the parameters of the R Bernoulli distributions (one for each annotator). In turn, $\mathbf{w}$ are the weights of a Logistic Regression model.

Following the generative process described above, we can now define $p(y_i^r|\mathbf{x}_i, z_i^r, \mathbf{w})$ as

$$p(y_i^r|\mathbf{x}_i, z_i^r, \mathbf{w}) = \Big(p_{\text{LogReg}}(y_i^r|\mathbf{x}_i, \mathbf{w})\Big)^{z_i^r} \Big(p_{\text{Rand}}(y_i^r|\mathbf{x}_i)\Big)^{1-z_i^r} \quad (5)$$

where $p_{\text{LogReg}}(y_i^r|\mathbf{x}_i, \mathbf{w})$ denotes the likelihood of the label provided by the r^{th} annotator for the instance $\mathbf{x}_i$ according to a multi-class Logistic Regression model with parameters $\mathbf{w}$, which for a classification task with K classes is given by

$$p_{\text{LogReg}}(y_i^r = k|\mathbf{x}_i, \mathbf{w}) = \frac{\exp(\mathbf{w}_k^T \mathbf{x}_i)}{\sum_{k'=1}^{K} \exp(\mathbf{w}_{k'}^T \mathbf{x}_i)}. \quad (6)$$

Similarly, $p_{\text{Rand}}(y_i^r|\mathbf{x}_i)$ denotes the likelihood of the label y_i^r according to a random model, which we assume to be uniformly distributed. Hence,

$$p_{\text{Rand}}(y_i^r = k|\mathbf{x}_i) = \frac{1}{K}. \quad (7)$$

To summarize, this is akin to saying that if $z_i^r = 1$ then the label provided by the r^{th} annotator (y_i^r) fits a Logistic Regression model, which is assumed to capture the correct (true) labeling process. Conversely, if $z_i^r = 0$ then y_i^r is assumed to be drawn from a random model where all the classes are equiprobable.

Since we do not actually observe the set $\mathcal{Z}$ we must treat the variables z_i^r as latent and marginalize them out of the likelihood by summing over all its possible outcomes. The (observed) data likelihood then becomes

$$p(\mathcal{D}|\theta) = \prod_{i=1}^{N} \prod_{r=1}^{R} \sum_{z_i^r \in \{0,1\}} p(z_i^r|\pi_r)\ p(y_i^r|\mathbf{x}_i, z_i^r, \mathbf{w}). \tag{8}$$

Making use of equations 3 and 5, this expression can be further simplified, giving

$$p(\mathcal{D}|\theta) = \prod_{i=1}^{N} \prod_{r=1}^{R} \Big(\pi_r\ p_{\text{LogReg}}(y_i^r|\mathbf{x}_i, \mathbf{w}) + (1-\pi_r)\ p_{\text{Rand}}(y_i^r|\mathbf{x}_i)\Big). \tag{9}$$

Our goal is then to estimate the maximum likelihood parameters θ_{ML}, which are found by determining $\theta_{\text{ML}} = \arg\max_\theta \ln p(\mathcal{D}|\theta)$.

At this point, it is important to note that extending this approach to sequence labeling problems, or any kind of structured prediction problems in general, could be as simple as replacing in equation 5 the probabilities $p_{\text{LogReg}}(y_i^r|\mathbf{x}_i, \mathbf{w})$ and $p_{\text{Rand}}(y_i^r|\mathbf{x}_i)$ with their sequence labeling counterparts, which for $p_{\text{LogReg}}(\cdot)$ could be an Hidden Markov Model (HMM) or a Conditional Random Field (CRF), and updating the remaining equations accordingly.

4.2. Expectation-Maximization

As with other latent variable models, we rely on the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to optimize this otherwise intractable maximization problem. The EM algorithm is an iterative method for finding maximum likelihood solutions for probabilistic models with latent variables, and consist of two steps: the E-step and M-step. In the E-step the posterior distribution of the latent variables is computed based on the current model parameters. This posterior distribution is then used to estimate the new model parameters (M-step). These two steps are then iterated until convergence.

If we observed the complete dataset $\{\mathcal{D}, \mathcal{Z}\}$ then the loglikelihood function would simply take the form $\ln p(\mathcal{D}, \mathcal{Z}|\theta)$. Since we only have access to the "incomplete" dataset $\mathcal{D}$, our state of the knowledge about the values of $\mathcal{Z}$ (the reliabilities of the annotators) can be given by the posterior distribution $p(\mathcal{Z}|\mathcal{D}, \theta)$. Therefore, instead of the complete data loglikelihood, we consider its expected value under the posterior distribution of the latent variable $p(\mathcal{Z}|\mathcal{D}, \theta)$, which corresponds to the E-step of the EM algorithm. Hence, in the E-step we use the current parameter values θ^{old} to find the posterior distribution of the latent variables in $\mathcal{Z}$. We then use this posterior distribution to find the expectation of the complete-data loglikelihood evaluated for some general parameter values θ. This expectation is given by

$$\begin{aligned}
&\mathbb{E}_{p(\mathcal{Z}|\mathcal{D},\theta_{old})}\Big[\ln p(\mathcal{D}, \mathcal{Z}|\theta)\Big] \\
&\quad = \sum_{\mathcal{Z}} p(\mathcal{Z}|\mathcal{D}, \theta_{old}) \ln p(\mathcal{D}, \mathcal{Z}|\theta) \\
&\quad = \sum_{i=1}^{N} \sum_{r=1}^{R} \sum_{z_i^r \in \{0,1\}} p(z_i^r|y_i^r, \mathbf{x}_i, \theta_{old}) \ln \Big(p(z_i^r|\pi_r)\ p(y_i^r|\mathbf{x}_i, z_i^r, \mathbf{w})\Big). \qquad (10)
\end{aligned}$$

The posterior distribution of the latent variables z_i^r (denoted by $\gamma(z_i^r)$)

can be estimated using the Bayes theorem giving

$$\begin{aligned}\gamma(z_i^r) &= p(z_i^r = 1 | y_i^r, \mathbf{x}_i, \theta^{old}) \\ &= \frac{p(z_i^r = 1|\pi_r^{old})\ p(y_i^r|\mathbf{x}_i, z_i^r = 1, \mathbf{w}^{old})}{p(z_i^r = 1|\pi_r^{old})\ p(y_i^r|\mathbf{x}_i, z_i^r = 1, \mathbf{w}^{old}) + p(z_i^r = 0|\pi_r^{old})\ p(y_i^r|\mathbf{x}_i, z_i^r = 0, \mathbf{w})} \\ &= \frac{\pi_r^{old}\ p_{\text{LogReg}}(y_i^r|\mathbf{x}_i, \mathbf{w}^{old})}{\pi_r^{old}\ p_{\text{LogReg}}(y_i^r|\mathbf{x}_i, \mathbf{w}^{old}) + (1 - \pi_r^{old})\ p_{\text{Rand}}(y_i^r|\mathbf{x}_i)} \qquad (11)\end{aligned}$$

where we also made use of equations 3 and 5.

The expected value of the complete data loglikelihood then becomes

$$\begin{aligned}\mathbb{E}_{p(\mathcal{Z}|\mathcal{D},\theta_{old})}\Big[\ln p(\mathcal{D}, \mathcal{Z}|\theta)\Big] = \sum_{i=1}^{N}\sum_{r=1}^{R} \gamma(z_i^r) \ln \Big(\pi_r\ p_{\text{LogReg}}(y_i^r|\mathbf{x}_i, \mathbf{w})\Big) \\ + (1 - \gamma(z_i^r)) \ln \Big((1 - \pi_r)\ p_{\text{Rand}}(y_i^r|\mathbf{x}_i)\Big). \qquad (12)\end{aligned}$$

In the M-step of the EM algorithm we maximize this expectation with respect to the model parameters θ, obtaining new parameter values θ^{new} given by

$$\theta^{new} = \arg\max_{\theta} \mathbb{E}_{p(\mathcal{Z}|\mathcal{D},\theta_{old})}\Big[\ln p(\mathcal{D}, \mathcal{Z}|\theta)\Big]. \qquad (13)$$

The EM algorithm can then be summarized as follows:

E-step Compute the posterior distribution of the latent variables z_i^r by making use of equation 11.

M-step Estimate the new model parameters $\theta^{new} = \{\boldsymbol{\pi}^{new}, \mathbf{w}^{new}\}$ given by

$$\mathbf{w}^{new} = \arg\max_{\mathbf{w}} \sum_{i=1}^{N}\sum_{r=1}^{R} \gamma(z_i^r) \ln p_{\text{LogReg}}(y_i^r|\mathbf{x}_i, \mathbf{w}) \qquad (14)$$

$$\widehat{\mathcal{Y}}^{new} = \arg\max_{\widehat{\mathcal{Y}}} p_{\text{LogReg}}(\widehat{\mathcal{Y}}|\mathcal{X}, \mathbf{w}^{new}) \qquad (15)$$

$$\pi_r^{new} = accuracy_r = \frac{\#\{i : y_i^r = \widehat{y}_i\}}{N_r} \qquad (16)$$

where N_r denotes the number of instances labeled by annotator r. In order to optimize equation 14 we use limited-memory BFGS (Liu and Nocedal, 1989). The first order derivate is given by

$$\nabla_{\mathbf{w}} = \sum_{i=1}^{N} \sum_{r=1}^{R} \left(\gamma(z_i^r) \sum_{k=1}^{K} \left(t_{ik}^r - p_{\text{LogReg}}(y_i = k|\mathbf{x}_i, \mathbf{w}) \right) \mathbf{x}_i \mathbf{x}_i^T \right) \tag{17}$$

where $\mathbf{t}_i^r$ is a vector representation of y_i^r in a 1-of-K coding scheme, thus t_{ik}^r would be 1 when k corresponds to the label provided by the r^{th} annotator and 0 otherwise.

Notice that this is very similar to the typical training of a multi-class Logistic Regression model. However, in this case, the contributions of the labels provided by each annotator to the loglikelihood are being weighted by her reliability, or in other words, by how likely it is for her to be correct. This makes our proposed approach quite easy to implement in practice.

5. Experiments

The proposed Multiple-Annotator Logistic Regression (MA-LR)[2] model was evaluated using both multiple-annotator data with simulated annotators and data manually labelled using AMT. The model was compared with the multi-class extension of the model proposed by Raykar et al. (2009, 2010), which is a latent ground truth model, and with two majority voting baselines:

- Soft Majority Voting (MVsoft): this corresponds to a multi-class Logistic Regression model trained with the *soft* probabilistic labels resultant from the voting process.

[2]Source code is available at: http://amilab.dei.uc.pt/fmpr/malr.tar.gz

- Hard Majority Voting (MVhard): this corresponds to a multi-class Logistic Regression model trained with the most voted labels resultant from the voting process (i.e. the most voted class for a given instance gets "1" and the others get "0").

In all experiments the EM algorithm was initialized with majority voting.

5.1. Simulated annotators

With the purpose of comparing the presented approaches in different classification tasks we used six popular benchmark datasets from the UCI repository[3] - a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. Since these datasets do not have labels from multiple annotators, the latter were simulated from the ground truth using two different methods. The first method, denoted "label flips", consists in randomly flipping the label of an instance with a given uniform probability $p(flip)$ in order to simulate an annotator with an average reliability of $(1 - p(flip))$. The second method, referred to as "model noise", seeks simulating annotators that are more consistent in their opinions, and can be summarized as follows. First, a multi-class Logistic Regression model is trained on the original training set. Then, the resulting weights $\mathbf{w}$ are perturbed, such that the classifier consistently "fails" in a coherent fashion throughout the test set. To do so, the values of $\mathbf{w}$ are standardized, and then random "noise" is drawn from a Gaussian distribution with zero mean and σ^2 variance and added to the weights $\mathbf{w}$. These weights are then "unstandardized" (by reversing

[3]http://archive.ics.uci.edu/ml/index.html

Table 1: Details of the UCI datasets

Dataset	Num. Instances	Num. Features	Num. Classes
Annealing	798	38	6
Image Segmentation	2310	19	7
Ionosphere	351	34	2
Iris	150	4	3
Parkinson's	197	23	2
Wine	178	13	3

the standardization process previously used), and the modified multi-class Logistic Regression model is re-applied to the training set in order to simulate an annotator. The quality of this annotator will vary depending on the value of σ^2 used.

Since in practice each annotator only labels a small subset of all the instances in the dataset, we introduce another parameter in this annotator simulation process: the probability $p(label)$ of an annotator labeling an instance.

Table 1 describes the UCI datasets used in these experiments. Special care was taken in choosing datasets that correspond to real data and that were among the most popular ones in the repository and, consequently, among the Machine Learning community. Datasets that were overly unbalanced, i.e. with too many instances of some classes and very few instances of others, were avoided. Despite that, the selection process was random, which resulted in a rather heterogeneous collection of datasets: with different sizes, dimensionalities and number of classes.

Figures 3 and 4 show the results obtained using 5 simulated annotators with different reliabilities using distinct simulation methods: "label flips" and "model noise" respectively. Although not all the results (i.e. using both simulation methods on all the six datasets) are presented here, we note that the omitted results are similar to those shown. Hence, to avoid redundancy and preserve brevity, only a random subset of these are presented. All the experiments use 10-fold cross-validation. Due to the stochastic nature of the simulation process of the annotators, each experiment was repeated 30 times and the average results were collected. The plots on the left show the root mean squared error (RMSE) between the estimated annotators accuracies and their actual accuracies evaluated against the ground truth. The plots on the center and on the right show, respectively, the trainset and testset accuracies. Note that here, unlike in "typical" supervised learning tasks, trainset accuracy is quite important since it indicates how well the models are estimating the *unobserved* ground truth labels from the opinions of the multiple annotators.

From a general perspective on the results of figures 3 and 4 we can conclude that both methods for learning from multiple annotators (MA-LR and Raykar) tend to outperform the majority voting baselines under most conditions. Not surprisingly, as the value of $p(label)$, and consequently the average number of instances labeled by each annotator, decreases, both the trainset and testset accuracies of all the approaches decrease or stay roughly the same. As expected, a higher trainset accuracy usually translates in a higher testset accuracy and a better approximation of the annotators accuracies (i.e. lower

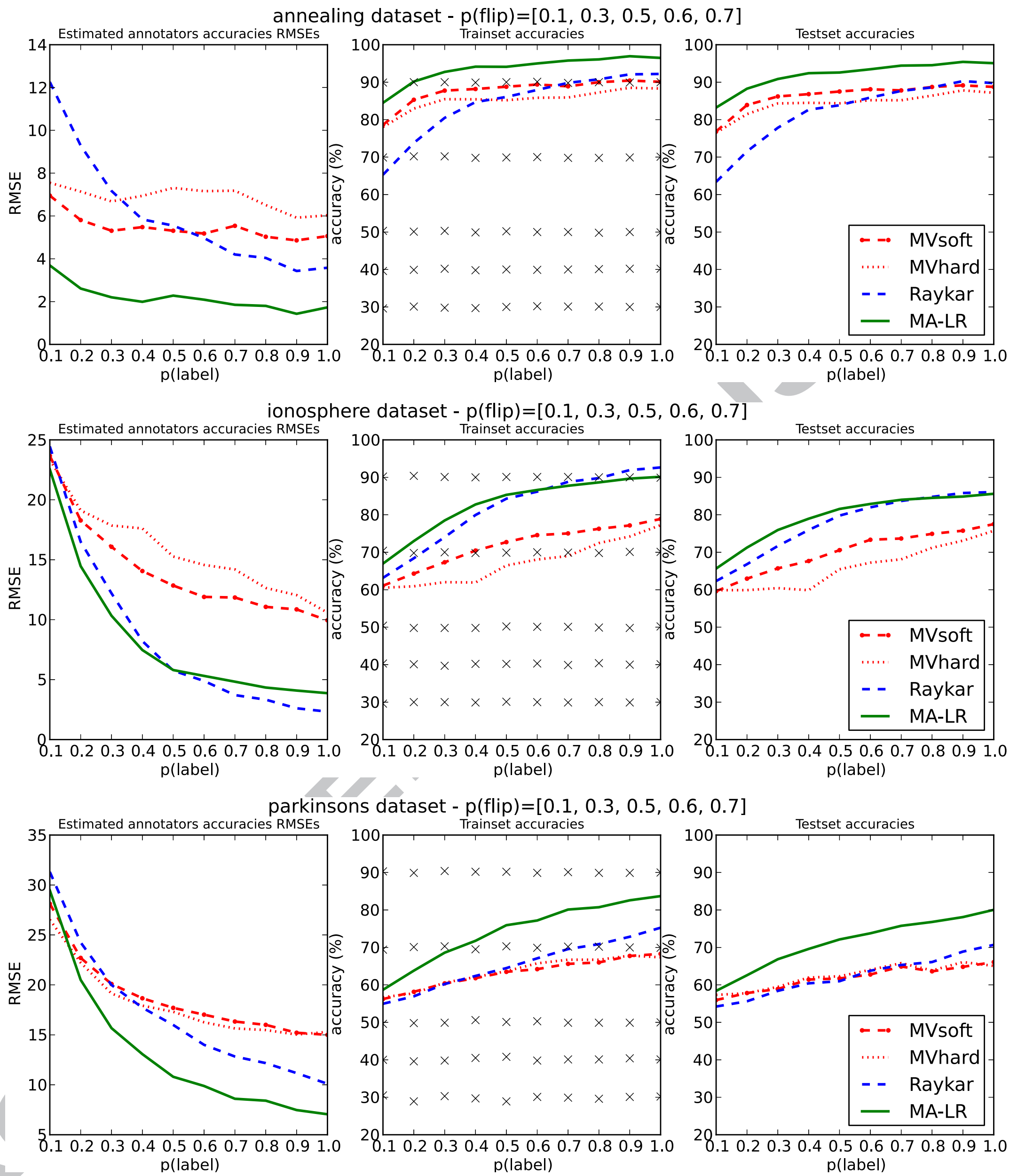

Figure 3: Results for the Annealing, Ionosphere and Parkinsons datasets using the "label flips" method for simulating annotators. The "x" marks indicate the average true accuracies of the simulated annotators.

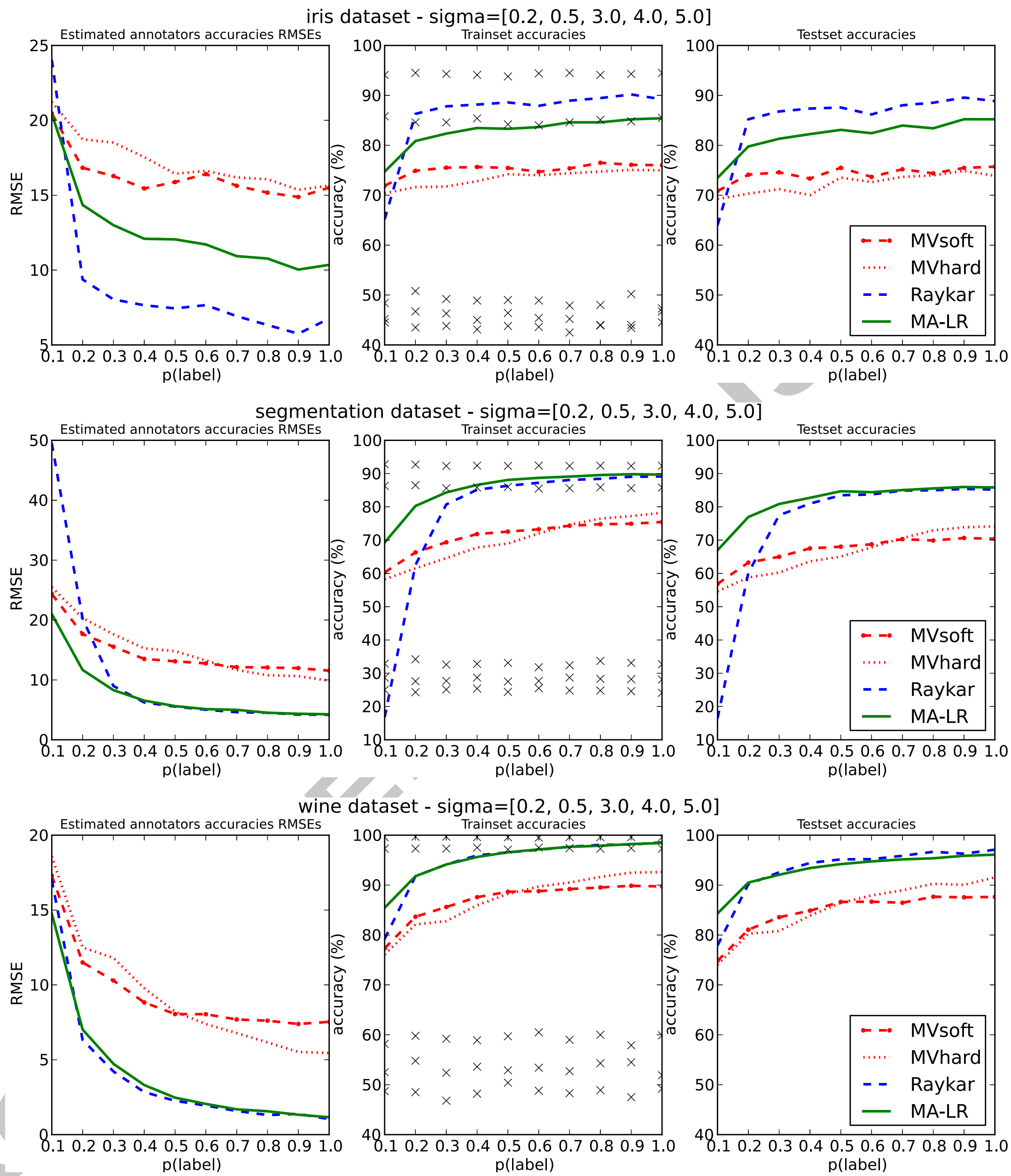

Figure 4: Results for the Iris, Segmentation and Wine datasets using the "model noise" method for simulating annotators. The "x" marks indicate the average true accuracies of the simulated annotators.

RMSE), since the approximation of the ground truth is also better.

A more careful analysis of the results reveals that, contrarily to the model by Raykar et al. (2009, 2010), the proposed model (MA-LR) is less prone to overfitting when the number of instances labeled by each annotator decreases. This is a direct consequence of the number of parameters used to model the annotators expertise. While the model by Raykar et al. (2009, 2010) uses a $K \times K$ confusion matrix for each annotator, making a total of RK^2 parameters, the proposed model only has R parameters. However, it is important to note that there is a tradeoff here, since the model by Raykar et al. can capture certain biases in the annotators answers by keeping a $K \times K$ confusion matrix for each annotator, which is not possible with the MA-LR model. Notwithstanding, in practice, on crowdsourcing platforms like AMT, the number of instances labeled by each annotator is usually low. Hence, we believe that the proposed model is preferable in most situations. Furthermore, our experimental results show that even when the number of instances labeled by each annotator is high, the MA-LR model can achieve similar or even better results than the model by Raykar et al. (2009, 2010).

5.2. Amazon Mechanical Turk

In order to assess the performance of the proposed model in learning from the labels of multiple non-expert human annotators and compare it with the other approaches, two experiments were conducted using AMT: sentiment polarity and music genre classification[4].

The sentiment polarity experiment was based on the sentiment analysis

[4]Datasets are available at: http://amilab.dei.uc.pt/fmpr/mturk-datasets.tar.gz

dataset introduced by Pang and Lee (2005), which corresponds to a collection of more than ten thousand sentences extracted from the movie review website RottenTomatoes[5]. These are labeled as positive or negative depending on whether they were marked as "fresh" or "rotten" respectively. From this collection, a random subset of 5000 sentences were selected and published on Amazon Mechanical Turk for annotation. Given the sentences, the workers were asked to provide the sentiment polarity (positive or negative). The remaining 5428 sentences were kept for evaluation.

For the music genre classification experiment, the audio dataset introduced by Tzanetakis and Cook (2002) was used. This dataset consists of a thousand samples of songs with 30 seconds of length and divided among 10 different music genres: classical, country, disco, hiphop, jazz, rock, blues, reggae, pop and metal. Each of the genres has 100 representative samples. A random 70/30 train/test split was performed on the dataset, and the 700 training samples were published on AMT for classification. In this case, the workers were required to listen to a 30-second audio excerpt and classify it as one of the 10 genres enumerated above.

On both experiments, the AMT workers were required to have an *HIT approval rate* - an AMT quality indicator that reflects the percentage of accepted answers of a worker - of 95%, which ensures some reliability on the quality of the answers.

Table 2 shows some statistics about the answers of the AMT workers for both datasets. Figure 5 further explore the distributions of the number of

[5]http://www.rottentomatoes.com/

Table 2: Statistics of the answers of the AMT workers for the two experiments performed. Note that the worker accuracies correspond to trainset accuracies.

	Sentiment polarity	Music genre
Number of answers collected	27747	2946
Number of workers	203	44
Avg. answers per worker (± std)	136.68 ± 345.37	66.93 ± 104.41
Min. answers per worker	5	2
Max. answers per worker	3993	368
Avg. worker accuracy (± std)	77.12 ± 17.10%	73.28 ± 24.16%
Min. worker accuracy	20%	6.8%
Max. worker accuracy	100%	100%

answers provided by each annotator and their accuracies for the sentiment polarity and music genre datasets. The figure reveals a highly skewed distribution of number of answers per worker, which support our intuition that on this kind of crowdsourcing platforms each worker tends to only provide a small number of answers, with only a couple of workers performing high quantities of labelings.

Standard preprocessing and features extraction techniques were performed on both experiments. In the case of the sentiment polarity dataset, the stop-words were removed and the remaining words were reduced to their root by applying a stemmer. This resulted in a vocabulary with size 8919, which still makes a bag-of-words representation computationally expensive. Hence, Latent Semantic Analysis (LSA) was used to further reduce the dimensionally

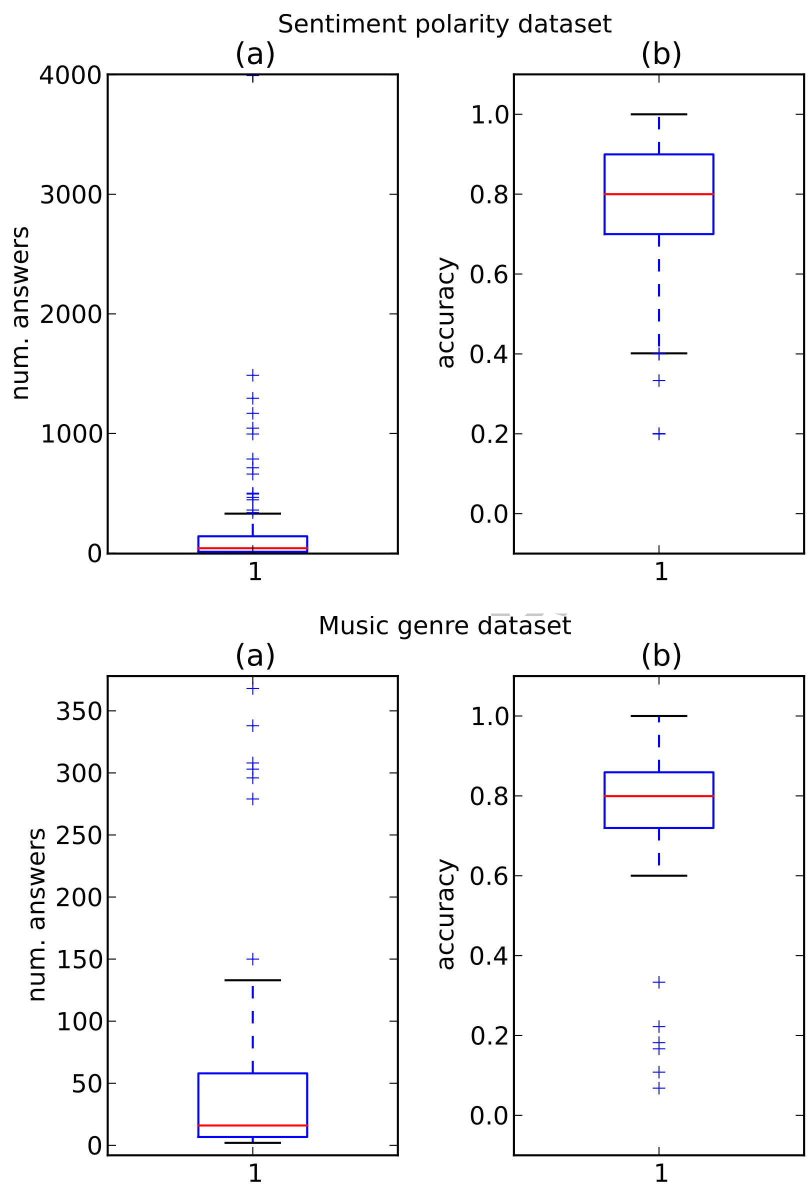

Figure 5: Boxplots for the number of answers (a) and for the accuracies (b) of the AMT workers for the sentiment polarity (top) and music genre (bottom) datasets.

Table 3: Trainset and testset accuracies for the different approaches on the datasets obtained from AMT.

	Sentiment polarity		Music genre	
Method	Train acc.	Test acc.	Train acc.	Test acc.
MVsoft	80.70%	71.65%	67.43%	60.33%
MVhard	79.68%	70.27%	67.71%	59.00%
Raykar	49.91%	48.67%	9.14%	12.00%
Raykar (w/prior)	84.92%	70.78%	71.86%	63.00%
MA-LR	85.40%	72.40%	72.00%	64.00%

of the dataset to 1200 features.

Regarding the music genre dataset, we used Marsyas[6], a standard music information retrieval tool, to extract a collection of commonly used features in this kind of tasks (Tzanetakis and Cook, 2002). These include means and variances of timbral features, time-domain Zero-Crossings, Spectral Centroid, Rolloff, Flux and Mel-Frequency Cepstral Coefficients (MFCC) over a texture window of 1 second. A total of 124 features were extracted. The details on these features fall out of the scope of this article. The interested reader is redirected to the appropriate literature (e.g. Aucouturier and Pachet (2003); Tzanetakis and Cook (2002)).

Table 3 presents the results obtained by the different methods on the sentiment polarity and music genre datasets. As expected, the results indicate that both annotator-aware methods are clearly superior when compared to

[6]http://marsyasweb.appspot.com

the majority voting baselines. Also, notice that due to the fact some annotators only label a very small portion of instances, the "standard" model by Raykar et al. (2009, 2010) performs very poorly (as bad as a random classifier) due to overfitting. In order to overcome this, a prior had to be imposed on the probability distribution that controls the quality of the annotators. In the case of the sentiment polarity task, a $Beta(1, 1)$ prior was used, and for the music genre task we applied a symmetric Dirichlet with parameter $\alpha = 1$. Despite the use of a prior, the model by Raykar et al. (2009, 2010) still performs worse than the proposed MA-LR model, which takes advantage of its single quality parameter per annotator to produce better estimates of the annotators' reliabilities. These results are coherent with our findings with the simulated annotators, which highlights the quality of the proposed model.

6. Conclusions and Future Work

In this paper we presented a new probabilistic model for supervised multi-class classification from multiple annotator data. Unlike previous approaches, in this model the reliabilities of the annotators are treated as latent variables. This design choice results in a model with various attractive characteristics, such as: its easy implementation and extension to other classifiers, the natural extension to structured prediction problems (as opposed to the commonly used latent ground truth models), and the ability to overcome the overfitting to which more complex models of the annotators expertise are susceptible as the number of instances labeled by each annotator decreases.

We empirically showed, using both simulated annotators and human-labeled data from Amazon Mechanical Turk, that under most conditions, the proposed approach achieves comparable or even better results when compared to a state of the art model (Raykar et al., 2009, 2010) despite its much smaller set of parameters to model the annotators expertise. In fact, it turned out that this reduced number of parameters plays a key role in making the model less prone to overfitting.

Future work will explore the behavior of the proposed model when we relax the assumption that the reliability of the annotators does not depend on the instances that they are labeling, similarly to what is done in Yan et al. (2010). Furthermore, the generalization to sequence labeling tasks will also be investigated.

References

Aucouturier, J., Pachet, F., 2003. Representing musical genre: A state of the art. Journal of New Music Research 32 (1), 83–93.

Bishop, C. M., 2006. Pattern Recognition and Machine Learning. Springer.

Dawid, A. P., Skene, A. M., 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. Journal of the Royal Statistical Society. Series C 28 (1), 20–28.

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, Series B 39 (1), 1–38.

Donmez, P., Schneider, J., Carbonell, J., 2010. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In: Proc. of the SIAM Int. Conf. on Data Mining. pp. 826–837.

Groot, P., Birlutiu, A., Heskes, T., 2011. Learning from multiple annotators with gaussian processes. In: Proc. of the 21st Int. Conf. on Artificial Neural Networks. Vol. 6792. pp. 159–164.

Howe, J., 2008. Crowdsourcing: Why the Power of the Crowd is Driving the Future of Business, 1st Edition. Crown Publishing Group, New York, NY, USA.

Jacobs, R., Jordan, M. I., Nowlan, S., Hinton, G., 1991. Adaptive mixtures of local experts. Neural Computation, 79–87.

Liu, D. C., Nocedal, J., 1989. On the limited memory BFGS method for large scale optimization. Mathematical Programming 45, 503–528.

Novotney, S., Callison-Burch, C., 2010. Cheap, fast and good enough: Automatic speech recognition with non-expert transcription. In: Proc. of the Human Language Technologies. Association for Computational Linguistics, pp. 207–215.

Pang, B., Lee, L., 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proc. of the ACL. pp. 115–124.

Raykar, V., Yu, S., Zhao, L., Jerebko, A., Florin, C., Valadez, G., Bogoni, L., Moy, L., 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In: Proc. of the 26th Int. Conf. on Machine Learning. pp. 889–896.

Raykar, V., Yu, S., Zhao, L., Valadez, G., Florin, C., Bogoni, L., Moy, L., 2010. Learning from crowds. Journal of Machine Learning Research, 1297–1322.

Smyth, P., Fayyad, U., Burl, M., Perona, P., Baldi, P., 1995. Inferring ground truth from subjective labelling of venus images. In: Advances in Neural Information Processing Systems. pp. 1085–1092.

Snow, R., O'Connor, B., Jurafsky, D., Ng, A., 2008. Cheap and fast - but is it good?: evaluating non-expert annotations for natural language tasks. In: Proc. of the Conf. on Empirical Methods in Natural Language Processing. pp. 254–263.

Surowiecki, J., 2004. The Wisdom of Crowds : Why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations. Doubleday.

Sutton, C., 2012. An introduction to conditional random fields. Foundations and Trends® in Machine Learning 4 (4), 267–373.

Tzanetakis, G., Cook, P., 2002. Musical genre classification of audio signals. Speech and Audio Processing, IEEE Transactions on 10 (5), 293–302.

Wu, O., Hu, W., Gao, J., 2011. Learning to rank under multiple annotators. In: Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence. pp. 1571–1576.

Yan, Y., Rosales, R., Fung, G., Dy, J., 2011. Active learning from crowds. In: Proc. of the 28th Int. Conf. on Machine Learning. pp. 1161–1168.

Yan, Y., Rosales, R., Fung, G., Schmidt, M., Valadez, G., Bogoni, L., Moy, L., Dy, J., 2010. Modeling annotator expertise: Learning when everybody knows a bit of something. Journal of Machine Learning Research 9, 932–939.

Highlights:

We propose a new probabilistic model for learning with multiple annotators.
The reliability of the different annotators is treated as a latent variable.
Model is able to achieve state of the art performance (or superior).
Reduced number of model parameters is able to avoid overfitting.
Model is easier to implement and extend to other classes of learning problems.