2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

# VARIATIONAL GAUSSIAN PROCESS FOR MISSING LABEL CROWDSOURCING CLASSIFICATION PROBLEMS

*Pablo Ruiz*[1*]*,Emre Besler*[2]*, Rafael Molina*[1]*, Aggelos K. Katsaggelos*[2]

[1] Dpto. de Ciencias de la Computación e I.A. Universidad de Granada.
[2] Dpt. of Electrical Engineering and Computer Science. Northwestern University.
e-mail: *mataran@decsai.ugr

## ABSTRACT

In this paper we address the crowdsourcing problem, where a classifier must be trained without knowing the real labels. For each sample, labels (which may not be the same) are provided by different annotators (usually with different degrees of expertise). The problem is formulated using Bayesian modeling, and considers scenarios where each annotator may label a subset of the training set samples only. Although Bayesian approaches have been previously proposed in the literature, we introduce Variational Bayes inference to develop an iterative algorithm where all latent variables are automatically estimated. In the experimental section the proposed model is evaluated and compared with other state-of-the-art methods on two real datasets.

***Index Terms***— Crowdsourcing, Gaussian process, multiple labels, variational inference, Bayesian modeling, classification, missing labels

## 1. INTRODUCTION

Supervised learning traditionally relies on a domain expert capable of providing the necessary supervision. The most common case is that of an expert providing annotations that serve as labels in classification problems. However, obtaining supervised data is usually very expensive in many real-life scenarios. In some cases, it is even impossible for annotators, no matter how knowledgeable, to come up with the true labels for all features.

With the recent developments in social media services, data can now be shared and processed by a large number of users. The use of labels from multiple annotators for data classification has become very popular especially after the proliferation of crowdsourcing services in the last decade. These labels are considered subjective, meaning that the expertise of the labelers and thus, the accuracy of the given labels cannot be guaranteed. However, using these subjective labels, an accurate estimate for the true labels can be obtained.

The term crowdsourcing was coined in 2006 by J. Howe [1] to describe the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call. Amazon Mechanical Turk [2] [3] (AMT) is an online system that allows the requesters to hire users from all over the world to perform crowdsourcing tasks. Services like AMT have made it practical to distribute labeling work to large groups of labelers and inexpensive to acquire labels in a short amount of time. As for another major example, Galaxy Zoo is a website where visitors label astronomical images and galaxies according to their morphological features. Very often, there is a lot of disagreement among the annotations.

Even though it is efficient to obtain the labels through crowdsourcing, there is a high probability that the collected labels are very noisy. This noise is caused by inconsistent labeling due to the fact that the reliability of annotators varies. Furthermore, some annotators might be considered experts for some parts of the data whereas their labels for other parts may be misleading. The higher the level of expertise of the annotators the fewer noisy labels s/he will generate. These issues bring up two main problems to be solved: First, how to estimate the final labels as accurately as possible. The accuracy here is determined by comparing the estimations with the ground truth, that may or may not be at hand. Second, how to estimate the expertise and reliability of the annotators. These problems are expected to be solved along with the training of a classifier using crowdsourcing data.

In this work, we propose a new Bayesian model that estimates the posterior distribution for each final label, along with two parameters that evaluate the reliability/expertise of each annotator. We extend the use of Variational Bayes inference for Gaussian Process (GP) classification to crowdsourcing problems. We show how the GP hyperparameters, the latent classifier and the parameters modelling each annotator's behavior can be estimated. We also describe how the model should be used to classify new samples. Then, we extend the

This work has been supported in part by the Ministerio de Economía y Competitividad under contract TIN2013-43880-R and the Department of Energy grant DE-NA0002520.

978-1-5090-0746-2/16/$31.00 ©2016 IEEE

proposed algorithm to the case where the labels of some annotators are missing for some part of the data.

The rest of paper is organized as follows. In section 2, a review of related works is presented along with the comparison with the model in this paper. The probabilistic modelling and inference procedure to estimate the posterior distributions of the variables and point estimates of the parameters are presented in sections 3 and 4, respectively. The classification rule for new samples is also provided in section 4. Experimental results are presented in section 5 and finally section 6 concludes the paper.

## 2. RELATED WORK

Multi-annotator data have been used in different contexts, as described next. To begin with, Dawid and Skeene [4] use multiple annotator data with conflicting labels to examine the error rates for medical data. They model the skills and biases of the annotators and estimate the model parameters using the EM algorithm. By casting the crowdsourcing problem as a utility optimization one subject to a budget constraint, Dommez and Carbonell [5] propose an approach to select the annotator and sample to be labeled. Iperiotis *et al.* [6] propose an algorithm for calculating the uncertainty of the labels and relabeling the samples whose labels are considered noisy. A variation on this problem is posed by Jin and Ghahramani [7], where a set of mutually exclusive labels are assigned to each sample and only one of the annotators has the correct label.

For crowdsourcing regression problems, Groot *et al.* [8] use a Gaussian Process (GP) modeling where the skill of each annotator is determined based on the noise of the observation process. For classification problems, Moreno *et al.* [9] use a hierarchical approach based on a Chinese restaurant process prior model that clusters the annotators into groups, combines the labels of all the annotators in a cluster, and then uses the cluster labels to learn the classifier. Liu *et al.* [10] introduce a probability distribution on the ability of each annotator. This ability corresponds to the probability of the true label and the one provided by the annotator being the same. The posterior distribution of the true label and abilities given the observed labels is then modeled and integrated over the abilities to perform Belief Propagation. The authors also use Mean Field inference on a related formulation and claim that their work constitutes the first application of variational inference to crowdsourcing. (See also [11], [12], which address the problem of minimizing the total price (i.e., redundancy) that must be paid to achieve a target overall reliability). Welinder *et al.* [13] propose a generative probabilistic model on a bird image dataset and use Bayesian inference to estimate both data difficulty and annotator bias.

The probabilistic formulation we follow in this paper is based on the one introduced in Raykar *et al.* [14] [15]. The authors use Logistic Regression (LR) to model the latent classifier and model, for a two-class problem, the distribution of the label provided by each annotator given the true one with two Bernoulli distribitions whose parameters are named specificity and sensibility. The EM algorithm is used to estimate all the unknowns. Yan *et al.* [16] [17] equate specificity and sensibility, dependent on the observed features and use LR to model these conditional distributions.

More recently, and following the approach formulated in [14] [15], Rodrigues *et al.* [18] have use the probit function on a GP prior to model the latent classifier and the specificity and sensibility defined in [14] [15]. Inference is performed using Expectation Propagation (EP), (see also Long *et al.* [19] for a similar formulation and inference procedure). In this paper we show the superiority of Variational Bayes (VB) inference over the EP methodology presented in [18] when the sigmoid function is used on GP prior realizations to model the latent classifier for the same observation procedure.

## 3. BAYESIAN MODELING

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ be a training set of $N$ samples with unknown labels $\mathbf{z} = (z_1, \ldots, z_N)^T \in \{0,1\}^N$. Let us assume that we have $R$ different annotators, and let $R_n \subseteq \{1, \ldots, R\}$ denote the subset of annotators who labeled the $n-th$ sample, and $N_r \subseteq \{1, \ldots, N\}$ the index subset of samples labeled by the $r-th$ annotator. Finally, let the set of labels provided by the $R$ different annotators be $\mathbf{Y} = \{y_n^r \in \{0,1\} \mid r \in R_n, n = 1, \ldots, N\}$. Our main goal is to infer the distribution of $\mathbf{z}$ given the information provided by the annotators.

To model the classification function relating each sample $\mathbf{x}_n$ to its corresponding hidden label $z_n$ we introduce a set of latent variables $\mathbf{f} = [f_1, \ldots, f_N]$ and write the conditional distribution of $\mathbf{z}$ given $\mathbf{f}$ as

$$\mathrm{p}(\mathbf{z}|\mathbf{f}) = \prod_{n=1}^{N} \left(\frac{1}{1+e^{-f_n}}\right)^{z_n} \left(\frac{e^{-f_n}}{1+e^{-f_n}}\right)^{1-z_n}. \quad (1)$$

For each sample, we have a Bernoulli distribution, where the two terms in the right hand side of the above equation are positive and add up to 1. When $\mathbf{x}_n$ belongs to class 1 (that is, $z_n$ = 1), only the first term is considered, and a very large positive value for $f_n$ is expected. When $\mathbf{x}_n$ belongs to class 0 (that is, $z_n$ = 0), only the second term is considered, and a very large negative value for $f_n$ is expected.

Then, given the features in $\mathbf{X}$ we model $\mathbf{f}$ using the following GP

$$\mathrm{p}(\mathbf{f}|\mathbf{X}, \boldsymbol{\Omega}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}), \quad (2)$$

where $\mathbf{K}$ is a kernel matrix which depends on $\mathbf{X}$ and a set of parameters $\boldsymbol{\Omega}$ to be estimated. We assume $\mathrm{p}(\boldsymbol{\Omega}) \propto$ const.

Following [14] we now define the following probabilities, named sensitivity and specificity, respectively, which relate the observed labels to the latent ones,

$$\alpha_r = \mathrm{p}(y^r = 1|z = 1), \quad \beta_r = \mathrm{p}(y^r = 0|z = 0). \quad (3)$$

Notice that other models for these conditional distributions, like the one in [16, 17] or a simplified model where $\alpha_r = \beta_r$ could also be used. Then, assuming that the annotators are independent, we have

$$\mathrm{p}(\mathbf{Y}|\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{r=1}^{R} \prod_{n \in N_r} \left[\alpha_r^{y_n^r}(1-\alpha_r)^{1-y_n^r}\right]^{z_n} \left[(1-\beta_r)^{y_n^r}\beta_r^{1-y_n^r}\right]^{1-z_n}, \tag{4}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_R)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_R)$.

Since $\alpha_j$ and $\beta_j$ with $j = 1, \ldots, R$ are probabilities (that is, they belong to $[0, 1]$), Beta hyperpriors are used to model them, i.e.,

$$\mathrm{p}(\boldsymbol{\alpha}) = \prod_{r=1}^{R} \mathrm{Beta}(\alpha_r|a_0^\alpha, b_0^\alpha),\ \mathrm{p}(\boldsymbol{\beta}) = \prod_{r=1}^{R} \mathrm{Beta}(\beta_r|a_0^\beta, b_0^\beta), \tag{5}$$

where we have removed the dependency on the parameters of the distribution for simplicity and

$$\mathrm{Beta}(\omega|a, b) \propto \omega^{a-1}(1-\omega)^{b-1}, \tag{6}$$

with

$$\mathrm{Mode}(\omega) = \frac{a-1}{a+b-2}. \tag{7}$$

Beta hyperpriors are usually used to model probabilities in Bayesian modeling (see [20]). Parameters $a$ and $b$ can be set to introduce prior information on $\omega$, and its uncertainty. Flat hyperpriors can be considered by using $a = b = 1$.

With the above information the probabilistic modelling of our crowdsourcing problem becomes

$$\mathrm{p}(\mathbf{Y}, \mathbf{z}, \mathbf{f}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}|\mathbf{X}) = \mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}|\mathbf{X}) = \mathrm{p}(\mathbf{Y}|\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta})\mathrm{p}(\mathbf{z}|\mathbf{f})\mathrm{p}(\mathbf{f}|\mathbf{X}, \boldsymbol{\Omega})\mathrm{p}(\boldsymbol{\alpha})\mathrm{p}(\boldsymbol{\beta})\mathrm{p}(\boldsymbol{\Omega}), \tag{8}$$

where $\boldsymbol{\Theta} = \{\mathbf{f}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}\}$.

## 4. VARIATIONAL INFERENCE

Our objective is to find the posterior distribution $\mathrm{p}(\boldsymbol{\Theta}|\mathbf{Y}, \mathbf{X}) = \mathrm{p}(\mathbf{Y}, \boldsymbol{\Theta}|\mathbf{X})/\mathrm{p}(\mathbf{Y}|\mathbf{X})$. However, it can only be approximated because $\mathrm{p}(\mathbf{Y}|\mathbf{X})$ can not be calculated.

We use the following approximation to the posterior distribution

$$\mathrm{q}(\boldsymbol{\Theta}) = \mathrm{q}(\mathbf{f})\mathrm{q}(\mathbf{z})\mathrm{q}(\boldsymbol{\alpha})\mathrm{q}(\boldsymbol{\beta})\mathrm{q}(\boldsymbol{\Omega}), \tag{9}$$

where $\mathrm{q}(\boldsymbol{\alpha})$, $\mathrm{q}(\boldsymbol{\beta})$ and $\mathrm{q}(\boldsymbol{\Omega})$ are degenerate distributions, that is, they take one value with probability one and the rest have probability zero, and $\mathrm{q}(\mathbf{f})$ and $\mathrm{q}(\mathbf{z})$ are non-degenerate.

We will find the approximating distribution $\mathrm{q}(\boldsymbol{\Theta})$ of $\mathrm{p}(\boldsymbol{\Theta}|\mathbf{Y}, \mathbf{X})$ by solving

$$\hat{\mathrm{q}}(\boldsymbol{\Theta}) = \arg\min_{\mathrm{q}(\boldsymbol{\Theta})} \mathrm{KL}(\mathrm{q}(\boldsymbol{\Theta})||\mathrm{p}(\boldsymbol{\Theta}|\mathbf{Y}, \mathbf{X})) = \arg\min_{\mathrm{q}(\boldsymbol{\Theta})} \int \mathrm{q}(\boldsymbol{\Theta}) \ln \frac{\mathrm{q}(\boldsymbol{\Theta})}{\mathrm{p}(\boldsymbol{\Theta}, \mathbf{Y}|\mathbf{X})} \mathrm{d}\boldsymbol{\Theta}. \tag{10}$$

The Kullback-Leibler (KL) divergence is always non-negative and it is equal to zero if and only if $\mathrm{q}(\boldsymbol{\Theta})$ and $\mathrm{p}(\boldsymbol{\Theta}, \mathbf{Y}|\mathbf{X})$ coincide. However, because of the functional form of (1), the KL divergence cannot be directly evaluated.

To overcome this problem, a variational bound [21] will be used. We have for any $\xi > 0$

$$\boldsymbol{\sigma}(f) = \frac{1}{1+e^{-f}} \geq \boldsymbol{\sigma}(\xi) \exp\left(\frac{f-\xi}{2} - \lambda(\xi)(f^2 - \xi^2)\right) \tag{11}$$

where $\lambda(\xi) = \frac{1}{2\xi}\left(\boldsymbol{\sigma}(\xi) - \frac{1}{2}\right)$. Thus, we have

$$\mathrm{p}(\mathbf{z}|\mathbf{f}) \geq \mathbf{H}(\mathbf{z}, \mathbf{f}, \boldsymbol{\xi}) = \prod_{n=1}^{N} \boldsymbol{\sigma}(\xi_n) \exp\left\{f_n(z_n - \frac{1}{2}) - \lambda(\xi_n)f_n^2 + \xi_n^2\lambda(\xi_n) - \frac{\xi_n}{2}\right\} \tag{12}$$

We then have the following lower bound for the joint distribution

$$\mathrm{p}(\boldsymbol{\Theta}, \mathbf{Y}|\mathbf{X}) \geq \mathbf{M}(\mathbf{f}, \mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}, \mathbf{Y}, \mathbf{X}, \boldsymbol{\xi}) = \mathrm{p}(\mathbf{Y}|\mathbf{z}, \boldsymbol{\alpha}, \boldsymbol{\beta})\mathbf{H}(\mathbf{z}, \mathbf{f}, \boldsymbol{\xi})\mathrm{p}(\mathbf{f}|\mathbf{X}, \boldsymbol{\Omega})\mathrm{p}(\boldsymbol{\alpha})\mathrm{p}(\boldsymbol{\beta})\mathrm{p}(\boldsymbol{\Omega}) \tag{13}$$

which produces

$$\mathrm{KL}(\mathrm{q}(\boldsymbol{\Theta})||\mathrm{p}(\boldsymbol{\Theta}|\mathbf{Y}, \mathbf{X})) \leq \mathrm{KL}(\mathrm{q}(\boldsymbol{\Theta})||\mathbf{M}(\boldsymbol{\Theta}, \mathbf{Y}, \mathbf{X}, \boldsymbol{\xi})), \tag{14}$$

and which is now mathematically tractable.

Now, we can use $\ln \mathbf{M}(\boldsymbol{\Theta}, \mathbf{Y}, \mathbf{X}, \boldsymbol{\xi})$ to obtain the best posterior distribution approximation, $\mathrm{q}(\boldsymbol{\Theta})$. This distribution consists of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\Omega}$, the values at which $\mathrm{q}(\boldsymbol{\alpha})$, $\mathrm{q}(\boldsymbol{\beta})$ and $\mathrm{q}(\boldsymbol{\Omega})$ degenerate, and the posterior distribution approximations $\mathrm{q}(\mathbf{f})$ and $\mathrm{q}(\mathbf{z})$.

For $\theta \in \boldsymbol{\Theta}$, let us denote by $\boldsymbol{\Theta}_\theta = \boldsymbol{\Theta} \backslash \theta$, the set $\boldsymbol{\Theta}$ with the exclusion of $\theta$, and $\mathrm{q}(\boldsymbol{\Theta}_\theta) = \prod_{\eta \in \boldsymbol{\Theta}_\theta} \mathrm{p}(\eta)$. For $\theta \in \{\mathbf{f}, \mathbf{z}\}$, the best approximating distribution, $\mathrm{q}(\theta)$, in the KL sense is given by

$$\ln \mathrm{q}(\theta) = <\ln \mathbf{M}(\boldsymbol{\Theta}, \mathbf{Y}, \mathbf{X}, \boldsymbol{\xi})>_{\mathrm{q}(\boldsymbol{\Theta}_\theta)} + \text{const.} \tag{15}$$

For $\mathrm{q}(\mathbf{f})$ we observe that $<\ln \mathbf{M}(\boldsymbol{\Theta}, \mathbf{Y}, \mathbf{X}, \boldsymbol{\xi})>_{\mathrm{q}(\boldsymbol{\Theta}_\mathbf{f})}$ is a quadratic function of $\mathbf{f}$ and so, the posterior distribution will be Gaussian with parameters:

$$\boldsymbol{\mu}_\mathbf{f} = \boldsymbol{\Sigma}_\mathbf{f}\left(<\mathbf{z}> - \frac{1}{2}\mathbf{1}\right), \tag{16}$$

$$\boldsymbol{\Sigma}_\mathbf{f} = \mathbf{K} - \mathbf{K}\mathbf{W}(\mathbf{I} + \mathbf{W}\mathbf{K}\mathbf{W})^{-1}\mathbf{W}\mathbf{K}, \tag{17}$$

where $\mathbf{W} = \sqrt{2}\boldsymbol{\Lambda}^{1/2}$, with $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda(\xi_1), \ldots, \lambda(\xi_N))$.

For $\mathrm{q}(\mathbf{z})$, each $z_n$ can only take two values. We have therefore

$$\mathrm{q}(z_n = 0) \propto \prod_{r \in R_n} (1-\beta_r)^{y_n^r}\beta_r^{1-y_n^r}, \tag{18}$$

$$\mathrm{q}(z_n = 1) \propto \exp(<f_n>) \prod_{r \in R_n} \alpha_r^{y_n^r}(1-\alpha_r)^{1-y_n^r}. \tag{19}$$

**Algorithm 1** GP for Crowdsourcing

**Require:** $\mathbf{X}$, $\mathbf{Y}$, $\boldsymbol{\xi}^0 = \mathbf{1}$, $\mathrm{q}^0(\mathbf{z})$, a product of Bernoulli distributions.

1: $k = 0$;
2: **repeat**
3: Calculate $\boldsymbol{\Omega}^{k+1}$ the minimizer of eq. (22) using $\mathrm{q}^k(\mathbf{z})$ and $\boldsymbol{\xi}^k$ in the function definition;
4: Calculate $\boldsymbol{\alpha}^{k+1}$ using $\mathrm{q}^k(\mathbf{z})$ in the rhs of eq. (20) ;
5: Calculate $\boldsymbol{\beta}^{k+1}$ using $\mathrm{q}^k(\mathbf{z})$ in the rhs of eq. (21) ;
6: Calculate $\mathrm{q}^{k+1}(\mathbf{f})$ using $\mathrm{q}^k(\mathbf{z})$, $\boldsymbol{\xi}^k$, and $\boldsymbol{\Omega}^{k+1}$ in the rhs of eqs. (16) and (17)
7: Calculate $\mathrm{q}^{k+1}(\mathbf{z})$ using $\boldsymbol{\alpha}^{k+1}$, $\boldsymbol{\beta}^{k+1}$, and $\mathrm{q}^{k+1}(\mathbf{f})$ in the rhs of eqs. (18) and (19);
8: Calculate $\boldsymbol{\xi}^{k+1}$ using $\mathrm{q}^{k+1}(\mathbf{f})$ in the rhs of eq. (23);
9: $k = k + 1$;
10: **until** Convergence
11: **output** $\mathrm{q}(\boldsymbol{\Theta})$

For $\theta \in \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Omega}\}$, the value where the distribution $\mathrm{q}(\theta)$ degenerates is obtained by maximizing $< \ln \mathbf{M}(\boldsymbol{\Theta}, \mathbf{Y}, \mathbf{X}, \boldsymbol{\xi}) >_{\mathrm{q}(\mathbf{f},\mathbf{z})}$ with respect to $\theta$. We obtain

$$\alpha_r = \frac{a_0^\alpha - 1 + \sum_{n \in N_r} < z_n > y_n^r}{a_0^\alpha + b_0^\alpha - 2 + \sum_{n \in N_r} < z_n >}, \quad r = 1, \ldots, R, \tag{20}$$

and analogously

$$\beta_r = \frac{a_0^\beta - 1 + \sum_{n \in N_r} (1 - < z_n >)(1 - y_n^r)}{a_0^\beta + b_0^\beta - 2 + \sum_{n \in N_r} (1 - < z_n >)} \quad r = 1, \ldots, R. \tag{21}$$

To estimate the kernel parameters we minimize the function

$$f(\boldsymbol{\Omega}) = \ln |\mathbf{K} + (2\boldsymbol{\Lambda})^{-1}| + \mathbf{u}^T (\mathbf{K} + (2\boldsymbol{\Lambda})^{-1})^{-1} \mathbf{u} \tag{22}$$

where $\mathbf{u} = 1/2 \times \boldsymbol{\Lambda}^{-1}(<\mathbf{z}> - \frac{1}{2}\mathbf{1})$ and, as we have already indicated, $\mathbf{K}$ depends on $\boldsymbol{\Omega}$.

Finally, to find $\boldsymbol{\xi}$ we maximize $< \ln \mathbf{M}(\boldsymbol{\Theta}, \mathbf{Y}, \mathbf{X}, \boldsymbol{\xi}) >_{\mathrm{q}(\boldsymbol{\Theta})}$ with respect to each $\xi_n$, which produces

$$\xi_n = \sqrt{< f_n >^2 + \boldsymbol{\Sigma}_{\mathbf{f}}(n, n)}. \tag{23}$$

The whole estimation procedure is summarized in Algorithm. 1.

We now describe the process to classify a new feature vector. Given a new feature vector $\mathbf{x}_*$ and the corresponding latent variable $f_*$, the predictive distribution for class $\mathcal{C}_1$ given $\mathbf{x}_*$ will then be

$$\mathrm{p}(\mathcal{C}_1 | \mathbf{x}_*) = \int \boldsymbol{\sigma}(f_*) \mathrm{p}(f_* | \mathbf{Y}) \mathrm{d}f_* \tag{24}$$

To calculate this quantity we first notice that

$$\mathrm{p}(f_* | \mathbf{Y}) = \int_{\mathbf{f}} \mathrm{p}(f_* | \mathbf{f}) \mathrm{p}(\mathbf{f} | \mathbf{Y}) d\mathbf{f} \approx \int_{\mathbf{f}} \mathrm{p}(f_* | \mathbf{f}) \mathrm{q}(\mathbf{f}) \mathrm{d}\mathbf{f}. \tag{25}$$

Furthermore,

$$\begin{pmatrix} \mathbf{f} \\ f_* \end{pmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{h} \\ \mathbf{h}^T & c \end{bmatrix} \right) \tag{26}$$

where $\mathbf{h} = [k(\mathbf{x}_1, \mathbf{x}_*), k(\mathbf{x}_2, \mathbf{x}_*), \ldots, k(\mathbf{x}_N, \mathbf{x}_*)]^T$, $c = k(\mathbf{x}_*, \mathbf{x}_*)$ and we have removed $\boldsymbol{\Omega}$ for simplicity.

Then, from eq. (26) we have

$$\mathrm{p}(f_* | \mathbf{f}) = \mathcal{N}(f_* | \mathbf{h}^T \mathbf{K}^{-1} \mathbf{f}, c - \mathbf{h}^T \mathbf{K}^{-1} \mathbf{h}), \tag{27}$$

and furthermore

$$\mathrm{q}(\mathbf{f}) = \mathcal{N}(\mathbf{f} | \boldsymbol{\mu}_{\mathbf{f}}, \boldsymbol{\Sigma}_{\mathbf{f}}). \tag{28}$$

Combining the above two equations in eq. (25) we obtain

$$\mathrm{p}(f_* | \mathbf{Y}) \approx \mathcal{N}(f_* | a, b^2) \tag{29}$$

where

$$a = \mathbf{h}^T \mathbf{K}^{-1} \boldsymbol{\mu}_{\mathbf{f}} \tag{30}$$

$$b^2 = \mathbf{h}^T \mathbf{K}^{-1} \boldsymbol{\Sigma}_{\mathbf{f}} \mathbf{K}^{-1} \mathbf{h} + c - \mathbf{h}^T \mathbf{K}^{-1} \mathbf{h}. \tag{31}$$

We finally have

$$\mathrm{p}(\mathcal{C}_1 | \mathbf{x}_*) = \int \boldsymbol{\sigma}(f_*) \mathcal{N}(f_* | a, b^2) \mathrm{d}f_* \approx \boldsymbol{\sigma}(\kappa(b^2) a) \tag{32}$$

where $\kappa(b^2) = (1 + \pi b^2/8)^{-1/2}$, (see [21] eq. (4.153) for details).

Notice that a threshold, $0 \leq \gamma \leq 1$, should now be used on $\mathrm{p}(\mathcal{C}_1 | \mathbf{x}_*)$ to assign a new sample $\mathbf{x}_*$ to $\mathcal{C}_1$. If $\gamma = 1/2$ we only need to check whether $a \geq 0$.

## 5. EXPERIMENTS

In this section we evaluate the proposed method (hencefort referred to as VGPCR) on two different real crowdsourcing datasets: Sentence Polarity and Music Genre. For both datasets, preprocessing and feature extraction were carried out in Rodrigues *et al.* [22], where the datasets were also divided in training and test sets. To obtain crowdsourcing labels, training sets were made available in Amazon Mechanical Turk [2]. Both datasets can be downloaded from Rodrigues' website[1].

The real labels for both datasets are known, which allows us to measure the performance of crowdsourcing methods. We compare VGPCR with the state-of-the-art methods proposed in [14] (Raykar) and [18] (Rodrigues). We also include in the comparison a Gaussian Process Classifier trained with the real labels (GP-Gold), and a GP classifier trained using Majority Voting (GP-MV). All methods are compared using two measures: Area Under ROC curve (AUC) and Overall Accuracy (OA) which is calculated for $\mathrm{p}(\mathcal{C}_1 | x_*) \geq 1/2$ in eq. (32).

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

| Sentence | True Label |
|---|---|
| "An original gem about an obsession with time." "A poignant comedy that offers food for thought." | "positive" |
| "This is amusing for about three minutes." "I didn't laugh. I didn't smile. I survived." | "negative" |

**Table 1**. Examples of positive and negative samples in Sentence Polarity dataset [23].

| Methods | Training set | | Testing set | |
|---|---|---|---|---|
| | AUC | OA | AUC | OA |
| GP-Gold | 0.9130 | 0.8376 | 0.8037 | 0.7307 |
| GP-MV | 0.8706 | 0.7856 | 0.7963 | 0.7179 |
| Raykar | 0.9094 | 0.9066 | 0.7090 | 0.6815 |
| Rodrigues | 0.9411 | 0.8940 | 0.7808 | 0.7181 |
| VGPCR | 0.8860 | 0.8132 | **0.8001** | **0.7257** |

**Table 2**. Figures of merits for all compared methods in Sentence Polarity dataset.

### 5.1. Sentence Polarity dataset

The first dataset, Sentence Polarity [23], consists of 10427 sentences extracted from movie reviews in "Rotten Tomatoes" website[2]. The goal is to decide whether a sentence corresponds to a "positive" or "negative" review. In Table 1 we show four sentences in the dataset.

In "Rotten Tomatoes", each author of a review labels it as "fresh" or "rotten" depending on whether it is "positive" or "negative", respectively. These labels are the ground truth. They are used for evaluation purposes only.

The dataset is divided into training and testing sets, with 4999 and 5428 samples, respectively. Preprocessing and feature extraction procedures are described in [22]. They result in feature vectors with 1200 components.

Table 2 contains the figures of merits for all the compared methods. For the test samples, VGPCR obtained $0.8001$ and $0.7257$ AUC and OA, respectively, while for the training set, the values were $0.8860$ and $0.8132$, respectively.

As expected, since GP-Gold was trained with the real labels, it obtained the best results for the testing set, while GP-MV performed worse than the proposed method.

Raykar's method obtained the worst results for the test set. However, for the training set, it obtained $0.9094$ and $0.9066$ AUC and OA, respectively, almost $20\%$ better than for the testing set, which indicates that the model overfits. This is probably due to the use of a linear kernel on the feature vector with no regularization on the regressors. Finally, notice that for the testing set, Rodrigues' method obtained $0.7808$ and $0.7181$ AUC and OA, respectively, i.e., $1.93\%$ and $0.76\%$ respectively lower than the proposed method. However, AUC and OA figures were very high for the training set. Therefore, Rodrigues' method also seems to overfit the data.

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

| Methods | Training set | | Testing set | |
|---|---|---|---|---|
| | AUC | OA | AUC | OA |
| GP-Gold | 0.9769 | 0.9570 | 0.9510 | 0.9463 |
| GP-MV | 0.8553 | 0.9190 | 0.8556 | 0.9150 |
| Raykar | 0.9290 | 0.9574 | 0.8595 | 0.8983 |
| Rodrigues | 0.9430 | 0.9224 | 0.8822 | 0.8560 |
| VGPCR | 0.9107 | 0.9353 | **0.9069** | **0.9263** |

**Table 3**. Figures of merits for all compared methods in Music Genre dataset.

### 5.2. Music Genre dataset

For the second experiment, we use the Music Genre dataset [24], which consists of 1000 fragments (30 secs. length) of songs. The goal is to distinguish between 10 music genres: *classical, country, disco, hiphop, jazz, rock, blues, reggae, pop*, and *metal*. We use a *one-vs-all* strategy to address this multi-class classification problem.

The dataset contains 100 samples from each genre, which were randomly divided in 70 samples for training and 30 for testing.

To label the training set, each annotator listened to a subset of fragments and labeled them as one of the ten genres listed above. 2945 labels were provided by 44 different annotators.

For preprocessing and feature extraction, the authors in [22] used Marsyas[3] music information tool, to extract 124 features from the original dataset. These features include: means and variances of timbral features, time-domain zero-crossings, spectral centroid, rolloff, flux and Mel-Frequency Cepstral Coefficients (MFCC).

In table 3 we show the results obtained by the compared methods on the Music Genre dataset. For the testing set, VGPCR obtained $0.9069$ and $0.9263$ AUC and OA, respectively. For the training set, the corresponding values were $0.9107$ and $0.9353$, respectively. These results are close to the obtained ones on the testing set, which means that no overfitting is produced.

Again, the results obtained by VGPCR are between those obtained by GP-MV and GP-Gold, however, the differences are more pronounced here than in the first experiment.

Raykar's and Rodrigues' methods obtained $0.8595$ and $0.8822$ AUC, respectively, for the testing set, which translates to $4.74\%$ and $2.47\%$ worse performance than VGPCR, respectively. Notice that, for the training set, both methods perform better than VGPCR, which means that they are capable to better fit the training samples, but the fitted models do not generalize well.

[3]http://marsyas.info/

## 6. CONCLUSION

In this paper we have addressed the crowdsourcing problem, where a classifier must be trained without knowing the real labels. Instead, a set of labels is provided by different annotators. The underlying classifier has been modeled using a Gaussian Process, and annotators were modeled with their corresponding sensitivity and sensibility values. Our formulation models scenarios where each annotator may label a subset of the training samples only. Variational Bayes inference has been used to derive an algorithm which allows us to estimate all the model parameters automatically. In the experimental section the proposed model has been evaluated and also compared with other state-of-the-art methods on two real datasets. The performed experiments indicate that the proposed method is more robust to overfitting, which leads to better results for new predictions.

## 7. REFERENCES


[1] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[2] R. Snow, B. O'Connor, D. Jurafsky, and A.Y. Ng, "Cheap and fast, but is it good?: evaluating non-expert annotations for natural language tasks," in *Proc. of EMNLP*, 2008, pp. 254–263.

[3] V.S. Sheng, F. Provost, and P.G. Ipeirotis, "Get another label? improving data quality and data mining using multiple, noisy labelers," in *Proc. of the 14th ACM SIGKDD*. 2008, pp. 614–622, ACM.

[4] A.P. Dawid and A.M. Skene, "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 20, 1979.

[5] P. Donmez and J.G. Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *Proc. of the 17th ACM Conf. on Information and knowledge management*. 2008, pp. 619–628, ACM.

[6] P.G. Ipeirotis, F. Provost, V.S. Sheng, and J. Wang, "Repeated labeling using multiple noisy labelers," *Data Mining and Knowledge Discovery*, vol. 28, no. 2, pp. 402–441, Mar. 2014.

[7] R. Jin and Z. Ghahramani, "Learning with multiple labels," in *Proc. of NIPS*, 2002, pp. 897–904.

[8] P. Groot, A. Birlutiu, and T. Heskes, "Learning from multiple annotators with Gaussian processes," in *Artificial Neural Networks and Machine Learning ICANN 2011*, pp. 159–164. Springer, 2011.

[9] P.G. Moreno, Y.W. Teh, F. Perez-Cruz, and A. Artés-Rodríguez, "Bayesian Nonparametric Crowdsourcing," *Journal of Machine Learning Research*, vol. 16, pp. 1607–1627, 2015.

[10] Qiang L., J. Peng, and A. Ihler, "Variational inference for crowdsourcing," in *Proc. of NIPS*, 2012, pp. 692–700.

[11] D.R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Proc. of NIPS*, pp. 1953–1961. 2011.

[12] D.R. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multi-class labeling," in *ACM SIGMETRICS Performance Evaluation Review*, 2013, vol. 41, pp. 81–92.

[13] P. Welinder, S. Branson, P. Perona, and S.J. Belongie, "The multidimensional wisdom of crowds," in *Proc. of NIPS*, 2010, pp. 2424–2432.

[14] V.C. Raykar, S. Yu, L.H. Zhao, G. Hermosillo Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *The Journal of Machine Learning Research*, vol. 11, pp. 1297–1322, 2010.

[15] V.C. Raykar, S. Yu, L.H. Zhao, A. Jerebko, C. Florin, G. Hermosillo Valadez, L. Bogoni, and L. Moy, "Supervised learning from multiple experts: whom to trust when everyone lies a bit," in *Proc. of the 26th Annual Int. Conf. on ML*. 2009, pp. 889–896, ACM.

[16] Y. Yan, R. Rosales, G. Fung, M.W. Schmidt, G. Hermosillo Valadez, L. Bogoni, L. Moy, and J.G. Dy, "Modeling annotator expertise: Learning when everybody knows a bit of something," in *Int. Conf. on Artificial Intelligence and Statistics*, 2010, pp. 932–939.

[17] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy, "Learning from multiple annotators with varying expertise," *Machine learning*, vol. 95, no. 3, pp. 291–327, 2014.

[18] F. Rodrigues, F. Pereira, and B. Ribeiro, "Gaussian process classification and active learning with multiple annotators," in *Proc. of ICML*, 2014, pp. 433–441.

[19] C. Long, G. Hua, and A. Kapoor, "A joint gaussian process model for active visual recognition with expertise estimation in crowdsourcing," *International Journal of Computer Vision*, vol. 116, no. 2, pp. 136–160, 2016.

[20] K.P. Murphy, *Machine Learning: A probabilistic Perspective*, The MIT Press, 2012.

[21] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag New York, NJ, USA, 2006.

[22] F. Rodrigues, F. Pereira, and B. Ribeiro, "Learning from multiple annotators: Distinguishing good from random labelers," *Pattern Recognition Letters*, vol. 34, no. 12, pp. 1428–1436, Sept. 2013.

[23] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. of ACL*, 2005, pp. 115–124.

[24] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. on Speech and Audio Process.*, vol. 10, no. 5, pp. 293–302, July 2002.