CLASSIFICATION OF MULTIPLE ANNOTATOR DATA USING VARIATIONAL GAUSSIAN PROCESS INFERENCE

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

[1] Dpt. of Electrical Engineering and Computer Science. Northwestern University.
[2] Dpto. de Ciencias de la Computación e I.A. Universidad de Granada.

e-mail: ⋆{emrebesler2020}@u.northwestern.edu,∗{mataran, rms}@decsai.ugr.es +aggk@eecs.northwestern.edu

ABSTRACT

In this paper we address supervised learning problems where, instead of having a single annotator who provides the ground truth, multiple annotators, usually with varying degrees of expertise, provide conflicting labels for the same sample. Once Gaussian Process classification has been adapted to this problem we propose and describe how Variational Bayes inference can be used to, given the observed labels, approximate the posterior distribution of the latent classifier and also estimate each annotator's reliability. In the experimental section, we evaluate the proposed method on both generated synthetic and real data, and compare it with state of the art crowdsourcing methods.

Index Terms— crowdsourcing, Gaussian process, multiple labels, variational inference, Bayesian modeling, classification.

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.

With the recent advent of social web services, data can now be shared and processed by a large number of users. The use of labels from multiple annotators for the classification of data has become a very popular approach especially after the proliferation of crowdsourcing services in the last decade. The term crowdsourcing was coined in 2006 by J. Howe 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 (AMT) is an online system that allows the requesters to hire users from all over the world to perform crowdsourcing tasks. Galaxy Zoo is a website where visitors label astronomical images. Computer Aided Diagnosis (CAD) systems are built from labels assigned by multiple experts who come from a diverse pool. Very often, there is a lot of disagreement among the annotations.

In this work, we extend the use of Variational Bayes (VB) 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.

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

The rest of paper is organized as follows. In section 2 a summary of related works is presented. 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

Although the multi-annotator data is a relatively new concept, it has been used for some time now. To begin with, Dawid and Skeene [1] used multiple annotator data with conflicting labels to examine the error rates for medical data. It is also constantly used for the repeated labeling approach [2] [3], that is based on determining the labels to be reacquired to improve the classification performance. A variation on this problem is posed by Jin and Ghahramani [4], where a set of mutually exclusive labels are assigned to each sample and only one of the annotators has the correct label.

After the usage of AMT has become more common, this field of research found its way to many different methods and applications. Groot et al [5] use Gaussian Processes for a crowdsourcing regression problem. Moreno et al [6] use a hierarchical approach that clusters the annotators into groups, combines the labels of all the annotators in a cluster, then uses the cluster labels to learn the classifier. Liu et al [7] transform the crowdsourcing problem into graphical models and then apply approximate variational methods like Belief Propagation and Mean Field alongside Expectation Maximization (EM). Karger et al [8] distribute the task among the annotators, that is, each annotator has part of the data to label; these parts overlap to create label redundancy for many samples. The problem is the optimization between redundancy cost and accuracy.

Raykar et al [9, 10] use Logistic Regression (LR) to estimate the latent classifier and relate the labels provided by each annotator to the latent classifier by defining conditional probabilities, named specificity and sensibility to be introduced later. The authors use EM to estimate all the unknowns. Yan et al [11, 12] make the conditional distribution of the observed labels given the true underlying ones dependent on the observed features and use LR to model these conditional distributions. Rodrigues et al [13] and Long et al [14] use a GP classifier for the latent classifier and use Expectation Propagation (EP) to learn all the model unknowns.

3. BAYESIAN MODELING

Let the training set be $\mathcal{D} = \{(\mathbf{x}_i, y_i^1, \ldots, y_i^R), i = 1, \ldots, N\}$, where $\mathbf{x}_i$ is a sample, N is the number of samples, R is the

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

number of annotators, and $y_i^j \in \{0,1\}$ denotes the label provided by the $j-$th annotator on the $i-$th sample. We denote by $\mathbf{y}^j = \{y_1^j, \ldots, y_N^j\}$ the labels provided by the $j-$th annotator, $\mathbf{y} = (y_1, \ldots, y_N)$ the corresponding true latent (hidden) labels, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, and $\mathbf{Y}^a = \{\mathbf{y}^1, \ldots, \mathbf{y}^R\}$. Our main goal is to infer the distribution of $\mathbf{y}$ given the information provided by the annotators.

To model the classification function relating each sample $\mathbf{x}_i$ to its corresponding hidden label y_i we follow a two stage procedure. Firstly, we introduce a set of latent variables $\mathbf{f} = [f_1, \ldots, f_N]$ and write the conditional distribution of $\mathbf{y}$ given $\mathbf{f}$ as

$$\mathrm{p}(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} \left(\frac{1}{1+e^{-f_i}}\right)^{y_i} \left(\frac{e^{-f_i}}{1+e^{-f_i}}\right)^{1-y_i}. \quad (1)$$

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 $\mathbf{x}_i$ belongs to class 1 (that is $y_i = 1$), only the first term is considered, and a very large positive value for f_i is expected. When $\mathbf{x}_i$ belongs to class 0 (that is $y_i = 0$), only the second term is considered, and a very large negative value for f_i is expected.

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

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

where $\mathbf{K}_{\boldsymbol{\Theta}}$ is a symmetric positive definite matrix, which is calculated using kernel functions (see [15] for details). These functions depend on a set of parameters $\boldsymbol{\Theta}$ which will be automatically estimated.

Following [9] we now define the following probabilities, named sensitivity and specificity, respectively, which relate the observed labels to the latent ones,

$$\alpha^j = \mathrm{p}(y^j = 1|y = 1) \quad (3)$$

$$\beta^j = \mathrm{p}(y^j = 0|y = 0) \quad (4)$$

Notice that other models for these conditional distributions are also possible. For instance, we could make them dependent on the features, as in [11, 12] or we could use a simplified model where $\alpha^j = \beta^j$. Then, assuming that the annotators are independent, we have

$$\mathrm{p}(\mathbf{Y}^a|\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{j=1}^{R} \left[\prod_{i=1}^{N} \left([\alpha^j]^{y_i^j}[1-\alpha^j]^{1-y_i^j}\right)^{y_i}\right] \times \left[\prod_{i=1}^{N} \left([1-\beta^j]^{y_i^j}[\beta^j]^{1-y_i^j}\right)^{1-y_i}\right] \quad (5)$$

where $\boldsymbol{\alpha} = (\alpha^1, \ldots, \alpha^R)$, $\boldsymbol{\beta} = (\beta^1, \ldots, \beta^R)$.

With all the above ingredients, the probabilistic modelling of our crowdsourcing problem becomes

$$\mathrm{p}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta}, \mathbf{f}, \mathbf{y}, \mathbf{Y}^a|\mathbf{X}) = \mathrm{p}(\boldsymbol{\alpha})\mathrm{p}(\boldsymbol{\beta})\mathrm{p}(\boldsymbol{\Theta})\mathrm{p}(\mathbf{f}|\mathbf{X}, \boldsymbol{\Theta}) \times \mathrm{p}(\mathbf{y}|\mathbf{f})\mathrm{p}(\mathbf{Y}^a|\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}). \quad (6)$$

We will use flat priors for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and also for $\boldsymbol{\Theta}$.

4. VARIATIONAL INFERENCE

Now we need to find $\mathrm{p}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta}, \mathbf{y}|\mathbf{Y}^a, \mathbf{X})$ which can only be approximated because $\mathrm{p}(\mathbf{Y}^a|\mathbf{X})$ can not be calculated.

Let $\boldsymbol{\Omega} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta}, \mathbf{f}, \mathbf{y}\}$. We use the following approximation to the posterior distribution

$$\mathrm{p}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta}, \mathbf{f}, \mathbf{y}|\mathbf{Y}^a, \mathbf{X}) \approx \mathrm{q}(\boldsymbol{\alpha})\mathrm{q}(\boldsymbol{\beta})\mathrm{q}(\boldsymbol{\Theta})\mathrm{q}(\mathbf{f})\mathrm{q}(\mathbf{y}) = \mathrm{q}(\boldsymbol{\Omega}) \quad (7)$$

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

We will find the approximating distribution by solving

$$\hat{\mathrm{q}}(\boldsymbol{\Omega}) = \arg\min_{\mathrm{q}(\boldsymbol{\Omega})} \mathrm{KL}(\mathrm{q}(\boldsymbol{\Omega})||\mathrm{p}(\boldsymbol{\Omega}|\mathbf{Y}^a, \mathbf{X})) = \arg\min_{\mathrm{q}(\boldsymbol{\Omega})} \int \mathrm{q}(\boldsymbol{\Omega}) \ln \frac{\mathrm{q}(\boldsymbol{\Omega})}{\mathrm{p}(\boldsymbol{\Omega}, \mathbf{Y}^a|\mathbf{X})} \mathrm{d}\boldsymbol{\Omega} \quad (8)$$

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

To overcome this problem, a variational bound [15] will be used. We have for any $\xi > 0$

$$\sigma(z) = \frac{1}{1+e^{-z}} \geq \sigma(\xi) \exp\left(\frac{z-\xi}{2} - \lambda(\xi)(z^2 - \xi^2)\right) \quad (9)$$

where

$$\lambda(\xi) = \frac{1}{2\xi}(\sigma(\xi) - \frac{1}{2}) \quad (10)$$

Thus, we have

$$\mathrm{p}(\mathbf{y}|\mathbf{f}) \geq \exp\left(\mathbf{y}^T\mathbf{f} - \mathbf{f}^T\boldsymbol{\Lambda}\mathbf{f} + \boldsymbol{\xi}^T\boldsymbol{\Lambda}\boldsymbol{\xi} - \frac{1}{2}\mathbf{1}^\mathbf{T}(\mathbf{f}+\boldsymbol{\xi})\right) \times \prod_{i=1}^{N} \sigma(\xi_i) = \mathbf{H}(\mathbf{y}, \mathbf{f}, \boldsymbol{\xi}) \quad (11)$$

where

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda(\xi_1), \lambda(\xi_2) \ldots \lambda(\xi_N)). \quad (12)$$

We then have the following lower bound for the joint distribution

$$\mathrm{p}(\boldsymbol{\Omega}, \mathbf{Y}^a|\mathbf{X}) \geq \mathbf{M}(\boldsymbol{\Omega}, \mathbf{Y}^a, \boldsymbol{\xi}|\mathbf{X}) = \mathrm{p}(\boldsymbol{\alpha})\mathrm{p}(\boldsymbol{\beta})\mathrm{p}(\boldsymbol{\Theta}) \times \mathrm{p}(\mathbf{f}|\mathbf{X}, \boldsymbol{\Theta})\mathbf{H}(\mathbf{y}, \mathbf{f}, \boldsymbol{\xi})\mathrm{p}(\mathbf{Y}^a|\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \quad (13)$$

which produces

$$\mathrm{KL}(\mathrm{q}(\boldsymbol{\Omega})||\mathrm{p}(\boldsymbol{\Omega}|\mathbf{Y}^a, \mathbf{X})) \leq \mathrm{KL}(\mathrm{q}(\boldsymbol{\Omega})||\mathbf{M}(\boldsymbol{\Omega}, \mathbf{Y}^a, \boldsymbol{\xi}|\mathbf{X})) \quad (14)$$

which is mathematically tractable.

Now, we can use $\ln \mathbf{M}(\boldsymbol{\Omega}, \mathbf{Y}^a, \boldsymbol{\xi}|\mathbf{X})$ to obtain $\hat{\mathrm{q}}(\boldsymbol{\Omega})$. This distribution consists of $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$, and $\hat{\boldsymbol{\Theta}}$, the values where $\hat{\mathrm{q}}(\boldsymbol{\alpha})$, $\hat{\mathrm{q}}(\boldsymbol{\beta})$, and $\hat{\mathrm{q}}(\boldsymbol{\Theta})$ are degenerate, and the posterior distributions approximations $\hat{\mathrm{q}}(\mathbf{f})$ and $\hat{\mathrm{q}}(\mathbf{y})$.

Let $\omega \in \boldsymbol{\Omega}$ and $\boldsymbol{\Omega}_\omega = \boldsymbol{\Omega} \backslash \omega$, then for $\omega \in \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Theta}\}$, if we fix $\mathrm{q}(\boldsymbol{\Omega}_\omega)$, the degenerate distributions minimizing the Kullback-Leibler divergence have the form

$$\mathrm{q}(\omega) = \begin{cases} 1 & \text{at } \omega_o = \arg\max_\omega < \ln \mathbf{M}(\boldsymbol{\Omega}, \mathbf{Y}^a, \boldsymbol{\xi}|\mathbf{X}) >_{\mathrm{q}(\boldsymbol{\Omega}_\omega)} \\ 0 & \text{elsewhere} \end{cases} \quad (15)$$

Furthermore for $\omega \in \{\mathbf{f}, \mathbf{y}\}$

$$\ln \mathrm{q}(\omega) = < \ln \mathbf{M}(\boldsymbol{\Omega}, \mathbf{Y}^a, \boldsymbol{\xi}|\mathbf{X}) >_{\mathrm{q}(\boldsymbol{\Omega}_\omega)} + \text{const} \quad (16)$$

Algorithm 1 GP for Crowdsourcing

Require: $\mathbf{X}$, $\mathbf{Y}^a$, $\boldsymbol{\xi}^0 = \mathbf{1}$, $\mathrm{q}^0(\mathbf{y})$ the product of Bernoulli distributions (we only need the probability of $y_i = 1$), an initial guess.

1: $n = 0$;
2: **repeat**
3: Calculate $\boldsymbol{\Theta}^{n+1}$ using $\mathrm{q}^n(\mathbf{y})$, $\boldsymbol{\xi}^n$ in eq. (23);
4: Calculate $\boldsymbol{\alpha}^{n+1}$ using $\mathrm{q}^n(\mathbf{y})$ in eq. (21);
5: Calculate $\boldsymbol{\beta}^{n+1}$ using $\mathrm{q}^n(\mathbf{y})$ in eq. (22);
6: Calculate $\mathrm{q}^{n+1}(\mathbf{f})$ using $\mathrm{q}^n(\mathbf{y})$, $\boldsymbol{\xi}^n$, $\boldsymbol{\Theta}^{n+1}$ in eq. (18);
7: Calculate $\mathrm{q}^{n+1}(\mathbf{y})$ using $\boldsymbol{\alpha}^{n+1}$, $\boldsymbol{\beta}^{n+1}$ and $\mathrm{q}^{n+1}(\mathbf{f})$ in eq. (19);
8: Calculate $\boldsymbol{\xi}^{n+1}$ using $\mathrm{q}^{n+1}(\mathbf{f})$ in eq. (20);
9: $n = n + 1$;
10: **until** Convergence

For $\mathrm{q}(\mathbf{f})$ we observe that $<\ln \mathbf{M}(\boldsymbol{\Omega}, \mathbf{Y}^a, \boldsymbol{\xi}|\mathbf{X})>_{\mathrm{q}(\boldsymbol{\Omega}_\mathbf{f})}$ is a quadratic function on $\mathbf{f}$ and so, the posterior distribution will be Gaussian. Mean and covariance matrix are calculated by taking first and second order derivatives of $\ln \mathrm{q}(\omega)$. Thus we obtain:

$$\boldsymbol{\mu}_\mathbf{f} = \boldsymbol{\Sigma}_\mathbf{f}(<\mathbf{y}> - \frac{1}{2}\mathbf{1}), \tag{17}$$

$$\boldsymbol{\Sigma}_\mathbf{f} = \mathbf{K}_{\boldsymbol{\Theta}} - \mathbf{K}_{\boldsymbol{\Theta}}\mathbf{W}(\mathbf{I} + \mathbf{W}\mathbf{K}_{\boldsymbol{\Theta}}\mathbf{W})^{-1}\mathbf{W}\mathbf{K}_{\boldsymbol{\Theta}} \tag{18}$$

where $\mathbf{W} = \sqrt{2}\boldsymbol{\Lambda}^{1/2}$.

For q(**y**), each y_i can only take two values. We have:

$$\mathrm{q}(y_i = 0) \propto \prod_{j=1}^{R}[1 - \beta_i]^{y_i^j}[\beta_i]^{1-y_i^j},$$

$$\mathrm{q}(y_i = 1) \propto \exp(<f_i>)\prod_{j=1}^{R}[\alpha_i]^{y_i^j}[1 - \alpha_i]^{1-y_i^j}, \tag{19}$$

We now proceed to find the values taken by the degenerate posterior distribution approximations.

To find $\boldsymbol{\xi}$ we have

$$\xi_i = \sqrt{<(f_i)^2>} = \sqrt{(<f_i>)^2 + \boldsymbol{\Sigma}_\mathbf{f}(i,i)}. \tag{20}$$

To find $\boldsymbol{\alpha}$ we again differentiate the bound and equate it to zero obtaining:

$$\alpha^j = \frac{\sum_i < y_i > y_i^j}{\sum_i < y_i >} \tag{21}$$

and analogously

$$\beta^j = \frac{\sum_i (1 - < y_i >)(1 - y_i^j)}{\sum_i (1 - < y_i >)}. \tag{22}$$

We finally proceed to estimate the kernel parameters. We have

$$\boldsymbol{\Theta}_o = \arg\min_{\boldsymbol{\Theta}} \ln|\mathbf{K}_{\boldsymbol{\Theta}} + (2\boldsymbol{\Lambda})^{-1}| + \mathbf{z}^T(\mathbf{K}_{\boldsymbol{\Theta}} + (2\boldsymbol{\Lambda})^{-1})^{-1}\mathbf{z} \tag{23}$$

where $\mathbf{z} = \frac{1}{2}\boldsymbol{\Lambda}^{-1}(<\mathbf{y}> - \frac{1}{2}\mathbf{1})$.

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

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

To calculate this quantity we first notice that

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

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 $\hat{\boldsymbol{\Theta}}$ for simplicity.

Then, from eq. (26)

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

furthermore

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

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

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

where

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

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

We finally have

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

where $\kappa(b^2) = (1 + \pi b^2/8)^{-1/2}$. (see [15] 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 $a \geq 0$. In the experimental section we will report ROC curves.

A simple multiclass extension is obtained by using a *one-vs-all* approach.

5. EXPERIMENTS

In this section we evaluate the proposed method on both synthetic and real datasets. We also compare with other the-state-of-the-art methods, such as Raykar et al [9], Yan et al[11, 12] and Rodrigues et al [13] which have been described in Section 2.

5.1. Synthetic Experiment

In Fig. 1 a) we plot the synthetic dataset. 200 samples are randomly selected in the interval $[-\pi, \pi]$. The real labels are assigned according to the sign of the cosine function on each sample, that is, if the cosine of a given sample is positive, the sample is assigned to class $\mathcal{C}_1$, but if the cosine is negative the sample is assigned to class $\mathcal{C}_0$.

We simulate 5 different annotators by fixing the values of sensitivity and specificity to $\boldsymbol{\alpha} = \{0.9, 0.7, 0.8, 0.1,\ 0.9\}$ and $\boldsymbol{\beta} = \{0.6, 0.8, 0.5, 0.2, 0.8\}$, respectively. If the true label of the i-th sample is $y_i = 1$, the j-th annotator assigns it to class $\mathcal{C}_1$ with probability α^j, while if the true label is $y_i = 0$, the annotator assigns it to class $\mathcal{C}_0$ with probability β^j. In Fig. 1 b)-f) we plot the labels assigned by each annotator. Notice that annotators 1,2,3, and

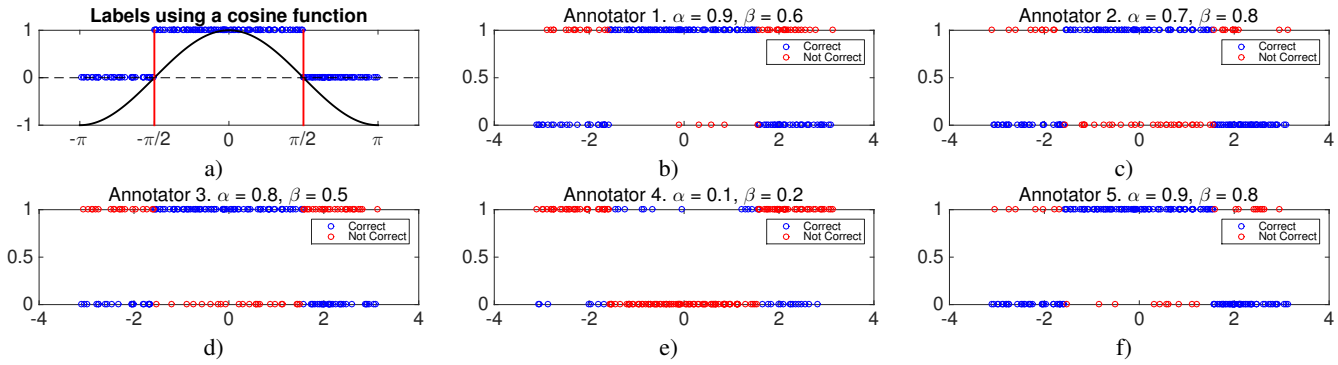

Fig. 1. a) Original data set labeled using sign of cosine function. b) - f) Labels provided by annotators 1,2,3,4 and 5 respectively.

Rea.	Raykar	Yan	Rodrigues	GPCR
1	0.4200	0.4200	1.0000	1.0000
2	0.4400	0.6640	0.9999	1.0000
3	0.5800	0.5800	1.0000	1.0000
4	0.5000	0.6825	1.0000	1.0000
5	0.4700	0.6781	0.9994	0.9998
6	0.4900	0.6711	1.0000	1.0000
7	0.4300	0.6637	0.9996	1.0000
8	0.5500	0.4500	1.0000	0.9999
9	0.4900	0.5100	0.9998	0.9999
10	0.5100	0.5100	1.0000	1.0000
Mean	0.4880	0.5829	0.9999	1.0000

Table 1. Area under ROC curve for 10 realizations of synthetic experiment.

5 make few mistakes; however annotator 4 is assigning most samples to the opposite class. This behavior is known in the literature as "spammer" [16].

The experiment is repeated 10 times with different training sets of 200 samples (100 of each class). In each realization, we also generate a test set with 200 samples (100 each class).

Table 1 shows the area under the ROC curve (AUC) for each realization for the compared methods. The proposed method, referred as GPCR (Gaussian Process for Crowdsourcing), manages to totally separate classes for most realizations, obtaining $AUC = 1.0$. The method proposed by Rodrigues et al also achieves high accuracy, reaching $AUC = 1.0$ in some cases; however we can see that the mean AUC is slightly worse than the proposed method. The methods of Raykar and Yan obtain AUC near 0.5, i.e., similar to random classifiers. In Fig. 1 a) we observe that $\mathcal{C}_0$ has two disconnected parts and $\mathcal{C}_1$ is in the middle. The methods of Raykar and Yan consider Logistic Regression to model the global classifier. Therefore their decision boundaries are hyperplanes which cannot separate the two classes with the given configuration. The proposed and Rodrigues' methods use GP to model the global classifier which uses the kernel trick [15] to define more complex decision boundaries than a hyperplane. In this case, we have used a Gaussian kernel (perhaps the most commonly used kernel in the literature) and as we have seen, it separates the two classes successfully. Sensitivity and specificity values obtained in realization 7 are shown in Table 2. We can see that all methods estimate values very near to the original ones. The highest error for the proposed method is 0.0594, while for Raykar and Rodrigues methods are 0.0834 and 0.0938, respectively.

	Original		Raykar		Rodrigues		GPCR	
Ann.	α	β	α	β	α	β	α	β
1	0.9	0.6	0.9482	0.5702	0.8789	0.5298	0.9594	0.6034
2	0.7	0.8	0.7169	0.7705	0.6435	0.7263	0.7070	0.7818
3	0.8	0.5	0.7731	0.5174	0.7365	0.4962	0.7685	0.5254
4	0.1	0.2	0.1187	0.2528	0.1886	0.2874	0.1117	0.2188
5	0.9	0.8	0.9834	0.8673	0.8062	0.7374	0.8993	0.8216

Table 2. Estimated sensitivity and specificity values in synthetic experiment.

5.2. Real Experiment

In this experiment, the proposed method is evaluated on a real dataset. This dataset is provided by Rodrigues in his website [13]. The dataset consists of more than 10000 sentences and the goal is to decide if they express a positive or negative sentiment. The dataset also contains the true labels (not available in a real case) which allow us to evaluate the performance of the different crowdsourcing methods.

The dataset is split into training (5000 samples) and testing (5428 samples) subsets. To obtain the labels from a set of annotators, the training set was made available on Amazon Mechanical Turk [17], where more than 27000 labels were obtained from 203 different annotators.

As can be observed from the total number of labels in the dataset, most of the samples are not labelled by all the annotators. To avoid this problem, we utilize here a reduced version of Rodrigues' training set. We consider only the two annotators with the maximum number of common labeled samples, we are currently working on the extension of our model to missing labels. Thus our training set has 946 samples and the labels are provided by only two annotators. To evaluate the performance, we use the original test set provided by Rodrigues *et al.* [13]

In Fig. 2, we plot the ROC curves generated by the compared methods. The dashed line is the ROC curve obtained by a GP classifier trained with the true training labels. This method reaches an $AUC = 0.7171$ and constitutes an upper bound for the proposed method. The green line is the ROC curve obtained by the proposed method which corresponds to $AUC = 0.7029$, very near to the upper bound. Red, cyan and blue lines are the ROC curves for Raykar,

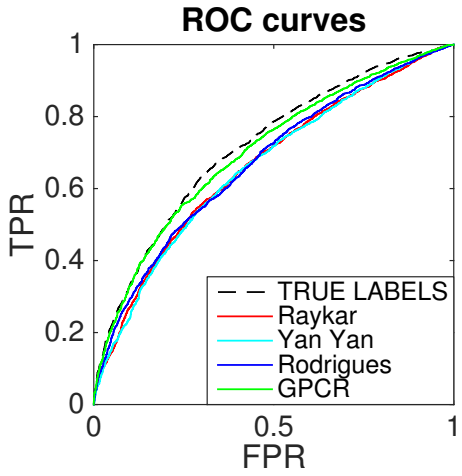

Fig. 2. ROC curves for experiment with real dataset.

Yan and Rodrigues methods, respectively. The corresponding areas under the ROC curves are $AUC = 0.6652$, $AUC = 0.6622$ and $AUC = 0.6723$, repectively. In all cases we observe that these methods perform slightly worse than the proposed method.

6. CONCLUSION

In this paper we present a new approach to address the crowdsourcing problem. The global classifier is modeled using Gaussian Processes, while the sensitivity and specificity of each annotator are modeled using two parameters to be estimated. Unlike other methods in the literature, the inference procedure is carried out using Variational Bayes inference, which leads to an iterative algorithm where all parameters are estimated automatically. In the experimental section we evaluate the performance of the proposed method with both synthetic and real datasets. We also compare with other state-of-the-art methods, and confirm that the proposed method outperforms them.

7. REFERENCES

[1] 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.

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

[3] 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.

[4] R. Jin and Z. Ghahramani, "Learning with multiple labels," in *Advances in neural information processing systems*, 2002, pp. 897–904.

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

[6] P. G. Moreno, Y. W. Teh, F. Perez-Cruz, and Antonio Artés-Rodríguez, "Bayesian Nonparametric Crowdsourcing," *arXiv preprint arXiv:1407.5017*, 2014.

[7] Q. Liu, J. Peng, and A. Ihler, "Variational inference for crowdsourcing," in *Advances in Neural Information Processing Systems*, 2012, pp. 692–700.

[8] 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, ACM.

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

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

[11] Y. Yan, R. Rosales, G. Fung, M. W. Schmidt, G. H. Valadez, L. Bogoni, L. Moy, and J. G. Dy, "Modeling annotator expertise: Learning when everybody knows a bit of something," in *International conference on artificial intelligence and statistics*, 2010, pp. 932–939.

[12] 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, June 2014.

[13] F. Rodrigues, F. Pereira, and B. Ribeiro, "Gaussian process classification and active learning with multiple annotators," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 433–441.

[14] 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, Jan. 2016.

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

[16] V. C. Raykar and S. Yu, "Ranking annotators for crowdsourced labeling tasks," in *Advances in neural information processing systems*, 2011, pp. 1809–1817.

[17] R. Snow, B. O'Connor, D. Jurafsky, and . Y. Ng, "Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2008, pp. 254–263.