Gaussian Process Classification and Active Learning with Multiple Annotators

Filipe Rodrigues FMPR@DEI.UC.PT
Centre for Informatics and Systems of the University of Coimbra (CISUC), 3030-290 Coimbra, PORTUGAL

Francisco C. Pereira CAMARA@SMART.MIT.EDU
Singapore-MIT Alliance for Research and Technology (SMART) 47 1 CREATE Way, SINGAPORE

Bernardete Ribeiro BRIBEIRO@DEI.UC.PT
Centre for Informatics and Systems of the University of Coimbra (CISUC), 3030-290 Coimbra, PORTUGAL

Abstract

Learning from multiple annotators took a valuable step towards modeling data that does not fit the usual single annotator setting, since multiple annotators sometimes offer varying degrees of expertise. When disagreements occur, the establishment of the correct label through trivial solutions such as majority voting may not be adequate, since without considering heterogeneity in the annotators, we risk generating a flawed model. In this paper, we generalize GP classification in order to account for multiple annotators with different levels expertise. By explicitly handling uncertainty, Gaussian processes (GPs) provide a natural framework for building proper multiple-annotator models. We empirically show that our model significantly outperforms other commonly used approaches, such as majority voting, without a significant increase in the computational cost of approximate Bayesian inference. Furthermore, an active learning methodology is proposed, which is able to reduce annotation cost even further.

1. Introduction

The problem of learning from multiple annotators occurs frequently in supervised learning tasks where, for diverse reasons such as cost or time, it is neither practical nor desirable to have a single annotator labeling all the data. With crowdsourcing (Howe, 2008) as a means for obtaining very large sets of labeled data, the problem of learning from multiple annotators is receiving increasing attention on behalf of researchers from various scientific communities, such as Speech, Music, Natural Language Processing, Computer Vision, etc. For many of these communities, the value of crowdsourcing platforms like Amazon Mechanical Turk (AMT)[1] and Crowdflower[2], has been empirically demonstrated. Concretely, it has been shown that, for many supervised learning tasks, the quality of the labels provided by multiple non-expert annotators can be as good as those of "experts" (Snow et al., 2008).

From a more general perspective, the concept of crowdsourcing goes much beyond dedicated platforms such as the AMT, and can often surface in more implicit ways. For example, the social web, where users' participation takes various forms, provides many interesting kinds of multi-annotator data (e.g. document tags, product ratings, user clicks, etc.).

Furthermore, the multiple annotators setting is not limited to the crowdsourcing phenomenon. For example, in the field of Medical Diagnosis, it is reasonable (and common) to have multiple "experts" providing their own opinions about whether or not an observable mass in a medical image is cancer, thereby avoiding the use of more invasive procedures (e.g., biopsy).

For this kind of problems, an obvious solution is to use majority voting. However, majority voting relies on the frequently wrong assumption that all annotators are equally reliable. Such an assumption is particularly threatening in more heterogeneous environments like AMT, where the reliability of the annotators can vary dramatically (Rodrigues et al., 2013a). It is therefore clear that targeted approaches for multiple-annotator settings are required. In fact, in the

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

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

last few years, many approaches have been proposed for the problem of supervised learning from multiple annotators. These span different kinds of problems, such as regression (Groot et al., 2011), classification (Raykar et al., 2010), sequence labeling (Rodrigues et al., 2013b), ranking (Wu et al., 2011), etc.

Despite the fact that crowdsourcing platforms like AMT provide researchers with a less expensive source of labeled data, for very large datasets the actual costs can still reach unacceptable amounts. This is specially true if we resort to repeated labeling (i.e. having the same instance labeled by multiple annotators) as a way to cope with the heterogeneity in the reliabilities of the annotators. Hence, ideally one would like to approach the problem of learning from multiple annotators in an active learning setting, thereby effectively reducing the annotation cost even further.

In this paper, we focus on classification problems, and generalize standard Gaussian process classifiers to explicitly handle multiple annotators with different levels of expertise. Gaussian processes (GPs) are flexible non-parametric Bayesian models that fit well within the probabilistic modeling framework (Barber, 2012). By explicitly handling uncertainty, GPs provide a natural framework for dealing with multiple annotators with different levels of expertise in a proper way. Furthermore, contrasting with previous works which usually rely on linear classifiers, we are bringing a powerful non-linear Bayesian classifier to multiple-annotator settings. Interestingly, it turns out that the computational cost of approximate Bayesian inference with Expectation Propagation (EP) involved in this new model is only greater up to a small factor (usually between 3 and 5) when compared with standard GP classifiers. Finally, GPs also provide a natural extension to active learning, thereby allowing us to choose the best instances to label and the best annotator to label them correctly in a simple and yet principled way, as we will demonstrate later.

2. State of the art

The problem of learning from multiple annotators has been around for quite some time, with the first notable early works being done by Dawid & Skene (1979). However, it was not until recently that the interest of the scientific community in the issue spiked, due to the massification of the social media and the Internet. As crowdsourcing platforms began getting the attention of researchers, new approaches for learning from from multiple annotators also started to appear. Raykar et al. (2010) proposed an approach for jointly learning the levels of expertise of different annotators and the parameters of a logistic regression classifier. The authors demonstrate that, by treating the unobserved true labels as latent variables, the proposed model significantly outperforms a standard logistic regression model trained on the majority voting labels. Yan et al. (2010) later extended this work to explicitly model the dependencies of annotators' labels on the instances they are labeling, and afterwards to active learning settings (Yan et al., 2011). Contrarily to these works, Welinder et al. (2010) approach the problem of learning from multiple annotators from a different perspective, and model each annotator as a multi-dimensional classifier in a feature space.

The problem of rating annotators according to their expertise is by itself a fundamental problem in the context of crowdsourcing. With that purpose, Liu & Wang (2012) extend the original work of Dawid & Skene (1979), where the annotators' expertise is modeled by means of a confusion matrix, by proposing a hierarchical Bayesian model, which allows each annotator to have her own confusion matrix, but at the same time regularizes these matrices through Bayesian shrinkage.

From a regression perspective, the problem of learning from multiple annotators has been addressed in the context of Gaussian processes by Groot et al. (2011). In their work, the authors assign different variances to the data points of the different annotators, thereby allowing them to have different noise levels, which are then automatically estimated by maximizing the marginal likelihood of the data.

On a different line of work, Bachrach et al. (2012) proposed a probabilistic graphical model that jointly models the difficulties of questions, the abilities of participants and the correct answers to questions in aptitude testing and crowdsourcing settings. By running approximate Bayesian inference with EP, the authors are able to query the model for the different variables of interest. Furthermore, by exploiting the principle of entropy, the authors devise an active learning scheme, which queries the answers which are more likely to reduce the uncertainty in the estimates of the model parameters. However, this work does not address the problem of explicitly learning a classifier from multiple-annotator data.

With respect to active learning applications with Gaussian processes, Lawrence et al. (2003) proposed a differential entropy score, which favours points whose inclusion leads to a large reduction in predictive (posterior) variance. This approach was then extended by Kapoor et al. (2007), by introducing a heuristic which balances posterior mean and posterior variance. The active learning methodology we propose further extends this work to multiple-annotator settings and introduces a new heuristic for selecting the best annotator to label an instance.

Annotation cost is an important issue in crowd labeling. Aiming at reducing this cost, Chen et al. (2013) consider the problem budget allocation in crowdsourcing environments, which they formulate as a Bayesian Markov Deci-

sion Process (MDP). In order to cope with computational tractability issues, the authors propose a new approximate policy to allocate a pre-fixed amount of budget among instance-worker pairs so that the overall accuracy can be maximized.

3. Approach

3.1. Gaussian process classification

In standard classification with Gaussian processes, given a set of N training input points $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$ and their corresponding class labels $\mathbf{y} = [y_1, ..., y_N]^T$, one would like to predict the class membership probability for a new test point $\mathbf{x}_*$. This can be achieved by using a latent function f, whose value is then mapped into the $[0, 1]$ interval by means of the probit function. For the particular case of binary classification, we shall adopt the convention that $y \in \{0, 1\}$, where 1 denotes the positive class, and 0 the negative. Hence, the class membership probability $p(y = 1|\mathbf{x})$ can be written as $\Phi(f(\mathbf{x}))$, where $\Phi(.)$ is the probit function. Gaussian process classification then proceeds by placing a GP prior over the latent function $f(\mathbf{x})$.

A GP (Rasmussen & Williams, 2005) is a stochastic process fully specified by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a positive definite covariance function $k(\mathbf{x}; \mathbf{x}') = \mathbb{V}[f(\mathbf{x}), f(\mathbf{x}')]$. In order to make predictions for a new test point $\mathbf{x}_*$, we first compute the distribution of the latent variable f_* corresponding to the test point $\mathbf{x}_*$

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{f})p(\mathbf{f}|\mathbf{X}, \mathbf{y})d\mathbf{f} \quad (1)$$

where $\mathbf{f} = [f_1, ..., f_N]^T$, and then use this distribution to compute the class membership distribution

$$p(y_* = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int \Phi(f_*)p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y})df_*. \quad (2)$$

3.2. Learning from multiple annotators

When learning from multiple annotators, instead of a single class label y_i for the i^{th} instance, we are given a vector of class labels $\mathbf{y}_i = [y_i^1, ..., y_i^R]$, corresponding to the noisy labels provided by the R annotators that labeled that instance.[3] Hence, a dataset $\mathcal{D}$ of size N is now defined as $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]^T$.

Let us now introduce a latent random variable z corresponding to the true class label for a given input point $\mathbf{x}$. When learning from multiple annotators, our goal is to estimate the posterior distribution of z_* for a new test point $\mathbf{x}_*$. Mathematically, we want to compute

$$p(z_* = 1|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int \Phi(f_*)p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})df_*. \quad (3)$$

The posterior distribution of the latent variable f_* is given by the following integral

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \int p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{f})p(\mathbf{f}|\mathbf{X}, \mathbf{Y})d\mathbf{f}. \quad (4)$$

By making use of Bayes rule, the joint posterior distribution of the latent variables $p(\mathbf{f}|\mathbf{X}, \mathbf{Y})$ becomes

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{f}|\mathbf{X})p(\mathbf{Y}|\mathbf{f})}{p(\mathbf{Y}|\mathbf{X})}$$

where the prior $p(\mathbf{f}|\mathbf{X})$ is a Gaussian distribution $\mathcal{N}(\mathbf{f}|\mathbf{m}, K)$ with some mean vector $\mathbf{m}$ (usually $\mathbf{m} = \mathbf{0}$) and covariance K obtained by evaluating the covariance function $k(\mathbf{x}; \mathbf{x}')$ between all input points, $p(\mathbf{Y}|\mathbf{f})$ is the likelihood term, and the denominator $p(\mathbf{Y}|\mathbf{X})$ corresponds to the marginal likelihood of the data.

So far, we have not established how to model $p(\mathbf{Y}|\mathbf{f})$. In order to do that, we make use of the latent variable z introduced earlier, which corresponds to the (latent) true class labels. Using this latent variable, we can define the data-generating process to be the following: for each input point $\mathbf{x}_i$ there is a (latent) true class label z_i, and the different R annotators then provide noisy versions y_i^r of z_i. This amounts to saying that $p(\mathbf{y}_i|f_i) = \sum_{z_i} p(z_i|f_i)p(\mathbf{y}_i|z_i)$. Assuming that the annotators make their decisions independently of each others allows $p(\mathbf{y}_i|z_i)$ to factorize, yielding

$$p(\mathbf{y}_i|f_i) = \sum_{z_i} p(z_i|f_i) \prod_{r=1}^{R} p(y_i^r|z_i)$$

where $p(z_i|f_i) = \Phi((-1)^{(1-z_i)} f_i)$ is the probit likelihood for values of $z_i \in \{0, 1\}$, and

$$y_i^r|z_i = 1 \sim Bernoulli(\alpha_r)$$
$$y_i^r|z_i = 0 \sim Bernoulli(1 - \beta_r).$$

The parameters of these Bernoullis, α_r and β_r, can therefore be interpreted as the sensitivity and specificity, respectively, of the r^{th} annotator.

Since the values of z are not observed, we have to marginalize over them by summing over all its possible values. Hence,

$$p(\mathbf{f}|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{f}|\mathbf{X}) \sum_{\mathbf{z}} p(\mathbf{Y}|\mathbf{z})p(\mathbf{z}|\mathbf{f})}{p(\mathbf{Y}|\mathbf{X})}$$

where we introduced the vector $\mathbf{z} = [z_1, ..., z_N]^T$.

[3] Notice that we are slightly changing notation from Section 3.1. Namely, vector $\mathbf{y}_i = [y_i^1, ..., y_i^R]$ is not to be confused with the vector $\mathbf{y} = [y_1, ..., y_N]^T$ of labels for the entire set of input points $\mathbf{X}$ used in equations 1 and 2.

By making use of the i.i.d. assumption of the data, we can re-write the posterior of the latent variables $\mathbf{f}$ as

$$p(\mathbf{f}|\mathbf{X},\mathbf{Y}) = \frac{1}{Z}p(\mathbf{f}|\mathbf{X})\prod_{i=1}^{N}\sum_{z_i\in\{0,1\}} p(\mathbf{y}_i|z_i)p(z_i|f_i) \quad (5)$$

where Z is a normalization constant corresponding to the marginal likelihood of the data $p(\mathbf{Y}|\mathbf{X})$. As with standard GP classification, the non-Gaussian likelihood term deems the posterior distribution of the latent variables $p(\mathbf{f}|\mathbf{X},\mathbf{Y})$ also non-Gaussian, thus making the integral in eq. 4 intractable. Hence, we proceed by approximating the posterior distribution of the latent variables $p(\mathbf{f}|\mathbf{X},\mathbf{Y})$ with a Gaussian distribution $q(\mathbf{f}|\mathbf{X},\mathbf{Y}) = \mathcal{N}(\boldsymbol{\mu},\Sigma)$ using Expectation Propagation (EP) (Minka, 2001).

In EP we approximate the likelihood by a local likelihood approximation in the form of an unnormalized Gaussian function in the latent variable f_i:

$$\sum_{z_i\in\{0,1\}} p(\mathbf{y}_i|z_i)p(z_i|f_i) \simeq t_i(f_i|\tilde{Z},\tilde{\mu}_i,\tilde{\sigma}_i^2) \triangleq \tilde{Z}_i\mathcal{N}(f_i|\tilde{\mu}_i,\tilde{\sigma}_i^2)$$

which defines the site parameters $\tilde{Z}$, $\tilde{\mu}_i$ and $\tilde{\sigma}_i^2$ of EP.

Also, in EP we abandon exact normalization for tractability. The product of the (independent) likelihoods t_i is then (Rasmussen & Williams, 2005):

$$\prod_{i=1}^{N} t_i(f_i|\tilde{Z},\tilde{\mu}_i,\tilde{\sigma}_i^2) = \mathcal{N}(\tilde{\boldsymbol{\mu}},\tilde{\Sigma})\prod_{i=1}^{N}\tilde{Z}_i$$

where $\tilde{\boldsymbol{\mu}}$ is a vector of $\tilde{\mu}_i$ and $\tilde{\Sigma}$ is a diagonal matrix with $\tilde{\Sigma}_{ii} = \tilde{\sigma}_i^2$.

The posterior $p(\mathbf{f}|\mathbf{X},\mathbf{Y})$ is then approximated by $q(\mathbf{f}|\mathbf{X},\mathbf{Y})$, which is given by

$$q(\mathbf{f}|\mathbf{X},\mathbf{Y}) \triangleq \frac{1}{Z_{EP}}p(\mathbf{f}|\mathbf{X})\prod_{i=1}^{N} t_i(f_i|\tilde{Z},\tilde{\mu}_i,\tilde{\sigma}_i^2) = \mathcal{N}(\boldsymbol{\mu},\Sigma) \quad (6)$$

with $\boldsymbol{\mu} = \Sigma\tilde{\Sigma}^{-1}\tilde{\boldsymbol{\mu}}$ and $\Sigma = (K^{-1}+\tilde{\Sigma}^{-1})^{-1}$. The normalization constant, $Z_{EP} = q(\mathbf{Y}|\mathbf{X})$, is the EP algorithm's approximation to the normalization term Z used in eq. 5.

All there is to do now, is to choose the parameters of the local approximating distributions t_i. In EP, this consists of four steps. In step 1, we compute the cavity distribution $q_{-i}(f_i)$ by dividing the approximate posterior marginal $q(f_i|\mathbf{X},\mathbf{Y}) = \mathcal{N}(f_i|\mu_i,\sigma_i)$ by the approximate likelihood term t_i, yielding

$$q_{-i}(f_i) \propto \int p(\mathbf{f}|\mathbf{X})\prod_{j\neq i}^{N} t_j(f_j,\tilde{Z}_j,\tilde{\mu}_j,\tilde{\sigma}_j^2)df_j \triangleq \mathcal{N}(f_i|\mu_{-i},\sigma_{-i}^2)$$

where

$$\mu_{-i} = \sigma_{-i}^2(\sigma_i^{-2}\mu_i - \tilde{\sigma}_i^{-2}\tilde{\mu}_i)$$
$$\sigma_{-i}^2 = (\sigma_i^{-2} - \tilde{\sigma}_i^{-2})^{-1}.$$

In step 2, we combine the cavity distribution with the exact likelihood term $\sum_{z_i\in\{0,1\}} p(\mathbf{y}_i|z_i)p(z_i|f_i)$ to get the desired (non-Gaussian) marginal, given by

$$\hat{q}(f_i) \triangleq \hat{Z}_i\mathcal{N}(\hat{\mu}_i,\hat{\sigma}_i^2) \simeq q_{-i}(f_i)\sum_{z_i\in\{0,1\}} p(\mathbf{y}_i|z_i)p(z_i|f_i).$$

By making use of the definitions of $p(\mathbf{y}_i|z_i)$ and $p(z_i|f_i)$ introduced earlier, this expression can be further manipulated, giving

$$\begin{aligned}\hat{q}(f_i) &\simeq q_{-i}(f_i)(1-\Phi(f_i))\prod_{r=1}^{R} p(y_i^r|z_i=0) \\ &\quad + q_{-i}(f_i)\Phi(f_i)\prod_{r=1}^{R} p(y_i^r|z_i=1) \\ &= b_i\mathcal{N}(f_i|\mu_{-i},\sigma_{-i}^2) + (a_i-b_i)\Phi(f_i)\mathcal{N}(f_i|\mu_{-i},\sigma_{-i}^2)\end{aligned} \quad (7)$$

where we defined

$$a_i = \prod_{r=1}^{R} p(y_i^r|z_i=1) = \prod_{r=1}^{R}(\alpha_r)^{(y_i)}(1-\alpha_r)^{(1-y_i)}$$
$$b_i = \prod_{r=1}^{R} p(y_i^r|z_i=0) = \prod_{r=1}^{R}(1-\beta_r)^{(y_i)}(\beta_r)^{(1-y_i)}.$$

Next, in the third step of EP, we choose a Gaussian approximation to the non-Gaussian marginal in eq. 7. This is done by moment matching. The derivation of the moments of eq. 7 is too extensive to be included here, hence we provide it as supplementary material[4], and show here only the results. The moments of eq. 7 are then given by

$$\hat{Z}_i = b_i + (a_i-b_i)\Phi(\eta_i)$$
$$\hat{\mu}_i = \mu_{-i} + \frac{(a_i-b_i)\sigma_{-i}^2\mathcal{N}(\eta_i)}{\Big[b_i+(a_i-b_i)\Phi(\eta_i)\Big]\sqrt{1+\sigma_{-i}^2}}$$
$$\hat{\sigma}_i^2 = \sigma_{-i}^2 - \frac{\sigma_{-i}^4}{1+\sigma_{-i}^2}\left(\frac{\eta_i\mathcal{N}(\eta_i)(a_i-b_i)}{b_i+(a_i-b_i)\Phi(\eta_i)} + \frac{\mathcal{N}(\eta_i)^2(a_i-b_i)^2}{(b_i+(a_i-b_i)\Phi(\eta_i))^2}\right)$$

where

[4] Supplementary material available at: http://amilab.dei.uc.pt/fmpr/publications/

$$\eta_i = \frac{\mu_{-i}}{\sqrt{1+\sigma^2_{-i}}}.$$

Finally, in step 4, we compute the approximations t_i that make the posterior have the desired marginals from step 3. Particularly, we want the product of the cavity distribution and the local approximation to have the desired moments, leading to (Rasmussen & Williams, 2005):

$$\begin{aligned}
\tilde{\mu}_i &= \tilde{\sigma}_i^2(\hat{\sigma}_i^{-2}\hat{\mu}_i - \sigma_{-i}^{-2}\mu_{-i}) \\
\tilde{\sigma}_i^2 &= (\hat{\sigma}_i^{-2} - \sigma_{-i}^{-2})^{-1} \\
\tilde{Z}_i &= \hat{Z}_i\sqrt{2\pi}\sqrt{\sigma^2_{-i}+\tilde{\sigma}_i^2}\exp\left(\frac{1}{2}\frac{\mu_{-i}-\tilde{\mu}_i}{\sigma^2_{-i}-\tilde{\sigma}_i^2}\right).
\end{aligned}$$

The different local approximating terms t_i are then updated sequentially by iterating through these four steps until convergence.

So far we have been assuming the annotators' parameters α_r and β_r to be given. However, we need to estimate those as well. This is done iteratively by scheduling the updates as follows: every n EP sweeps through the data, or alternatively, when the difference in the marginal likelihood between two consecutive iterations ϵ falls below a certain threshold[5], the values of α_r and β_r are re-estimated as:

$$\alpha_r = \frac{\sum_{i=1}^N y_i^r p(z_i=1|\mathbf{X},\mathbf{Y})}{\sum_{i=1}^N p(z_i=1|\mathbf{X},\mathbf{Y})} \tag{8}$$

(9)

$$\beta_r = \frac{\sum_{i=1}^N (1-y_i^r)(1-p(z_i=1|\mathbf{X},\mathbf{Y}))}{\sum_{i=1}^N 1-p(z_i=1|\mathbf{X},\mathbf{Y})}. \tag{10}$$

Although this will raise the computational cost of EP, as we shall see in Section 4, this increase is only by a small factor.

In order to make predictions, we make use of the EP approximation to the posterior distribution $q(\mathbf{f}|\mathbf{X},\mathbf{Y})$ defined in eq. 6, and plug it in eq. 4 to compute the predictive mean and variance of the latent variable f_*:

$$\begin{aligned}
\mathbb{E}_q[f_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y}] &= \mathbf{k}_*^T(K+\tilde{\Sigma})^{-1}\tilde{\boldsymbol{\mu}} \\
\mathbb{V}_q[f_*|\mathbf{x}_*,\mathbf{X},\mathbf{Y}] &= k(\mathbf{x}_*;\mathbf{x}_*) - \mathbf{k}_*^T(K+\tilde{\Sigma})^{-1}\mathbf{k}_*
\end{aligned}$$

where $\mathbf{k}_*$ is a vector whose entries correspond to the covariance function $k(\mathbf{x};\mathbf{x}')$ evaluated between the test point $\mathbf{x}_*$ and all the training input points.

Finally, the approximate predictive distribution for the true class label z_* is given by the integral in eq. 3, which can be analytically approximated as

$$\begin{aligned}
&q(z_*=1|\mathbf{x}_*,\mathbf{X},\mathbf{Y}) \\
&\quad = \Phi\left(\frac{\mathbf{k}_*^T(K+\tilde{\Sigma})^{-1}\tilde{\boldsymbol{\mu}}}{\sqrt{1+k(\mathbf{x}_*;\mathbf{x}_*)-\mathbf{k}_*^T(K+\tilde{\Sigma})^{-1}\mathbf{k}_*}}\right).
\end{aligned}$$

3.3. Active learning

The full Bayesian treatment of the Gaussian process framework provides natural extensions to active learning settings, which can ultimately reduce the annotation cost even further.

In active learning with multiple annotators our goal is twofold: (1) pick an instance to label next and (2) pick the best annotator to label it. For simplicity, we choose to treat the two problems separately. Hence, in order to pick an instance to label, we take the posterior distribution of the latent variable $p(f_u|\mathbf{x}_u,\mathbf{X},\mathbf{Y}) = \mathcal{N}(f_u|\mu_u,\sigma_u^2)$ for all unlabeled data points $\mathbf{x}_u \in \mathbf{X}_u$ and compute

$$\mathbf{x}^* = \arg\min_{\mathbf{x}_u\in\mathbf{X}_u} \frac{|\mu_u|}{\sqrt{1+\sigma_u}}. \tag{11}$$

This approach is analogous to the one proposed in (Kapoor et al., 2007) for single-annotator settings, and provides a balance between the distance to the decision boundary, given by the posterior mean $|\mu_u|$, and the posterior variance σ_u (uncertainty) associated with that point.

As for the choice of the annotator to label the instance picked, we proceed by identifying the annotator who is more likely to label it correctly given our current state of knowledge, i.e. given our prior beliefs of the class which the instance belongs to and the information about the levels of expertise of the different annotators. Mathematically, we want to pick an annotator r^* to maximize

$$\begin{aligned}
r^* &= \arg\max_r \Big[p(y^r=1|z=1)p(z=1|\mathbf{x}^*,\mathbf{X},\mathbf{Y}) \\
&\qquad + p(y^r=0|z=0)p(z=0|\mathbf{x}^*,\mathbf{X},\mathbf{Y})\Big] \\
&= \arg\max_r \Big[\alpha_r p(z=1|\mathbf{x}^*,\mathbf{X},\mathbf{Y}) \\
&\qquad + \beta_r(1-p(z=1|\mathbf{x}^*,\mathbf{X},\mathbf{Y}))\Big].
\end{aligned} \tag{12}$$

However, since we are now actively picking the annotators, there is a risk of generating a model that is biased towards labels from a single annotator when using this heuristic. This happens because if a single annotator provides the majority of the labels, the estimate of the ground truth will be biased towards her opinion. Consequently, her sensitivity and specificity parameters will also be biased, and she might end up being selected over and over. In order to address this issue, we introduce a dependency on the annotator r when estimating α_r and β_r. Namely, we replace $p(z_i=1|\mathbf{X},\mathbf{Y})$ with $p(z_i=1|\mathbf{X}\backslash\mathbf{x}^r,\mathbf{Y}\backslash\mathbf{y}^r)$ in equations 8

[5] During the experiments, these values were set to $n=3$ and $\epsilon = 10^{-4}$.

and 10, where $\mathbf{Y}\backslash\mathbf{y}^r$ denotes all the labels except the ones from annotator r, thereby deeming the ground truth estimates used for computing the reliability parameters α_r and β_r of annotator r, independent of her own answers.

4. Experiments

The proposed approaches[6] are validated using both real and simulated annotators on real datasets from different application domains.

4.1. Simulated annotators

In order to simulate annotators with different levels of expertise, we start by assigning a sensitivity α_r and specificity β_r to each of the simulated annotators. Then for each training point, we simulate the answer of the r^{th} annotator by sampling y_i^r from a $Bernoulli(\alpha_r)$ if the training point belongs to the positive class, and by sampling y_i^r from $Bernoulli(1-\beta_r)$ otherwise. This way, we can simulate annotators whose expected values for the sensitivity and specificity will tend to α_r and β_r respectively, as the number of training points goes to infinity.

This annotator simulation process is applied to various datasets from the UCI repository[7], and the results of the proposed approach (henceforward referred to as GPC-MA) is compared with two baselines: one consisting of using the majority vote for each instance (referred as GPC-MV), and another baseline consisting of using all data points from all annotators as training data (GPC-CONC). Note that if we simulate 7 annotators, then the dataset for the latter baseline will be 7 times larger than the former one. In order to also provide an upper bound/baseline we also show the results of a Gaussian process classifier applied to the true (golden) labels $\mathbf{z}$ (referred as GPC-GOLD).

Table 1 shows the results obtained in 6 UCI datasets, by simulating 7 annotators with sensitivities $\alpha = [0.9, 0.9, 0.8, 0.4, 0.3, 0.4, 0.6, 0.5]$ and specificities $\beta = [0.8, 0.9, 0.9, 0.4, 0.5, 0.5, 0.5, 0.4]$. For all experiments, a random 70/30% train/test split was performed and a isotropic squared exponential covariance function was used. Taking advantage of the stochastic nature of the annotators' simulation process, we repeat each experiment 30 times and always report the average results. Besides testset results, we also report performance metrics on the trainset because this corresponds to the important problem of uncovering the ground truth labels from the noisy answers of multiple annotators. We highlight in bold the highest performing method, excluding the upper-bound (GPC-GOLD).

[6]Source code and datasets are available at: http://amilab.dei.uc.pt/fmpr/software/

[7]http://archive.ics.uci.edu/ml/

		Trainset		Testset	
	Method	Acc.	AUC	Acc.	AUC
ionosphere	GPC-GOLD	1.000	1.000	0.900	0.999
	GPC-CONC	0.811	0.880	0.743	0.830
	GPC-MV	0.726	0.853	0.693	0.708
	GPC-MA	**0.978**	**0.998**	**0.889**	**0.987**
pima	GPC-GOLD	1.000	1.000	0.993	1.000
	GPC-CONC	0.848	0.900	0.860	0.930
	GPC-MV	0.840	0.955	0.860	0.967
	GPC-MA	**0.994**	**1.000**	**0.991**	**1.000**
parkinsons	GPC-GOLD	1.000	1.000	0.992	0.999
	GPC-CONC	0.827	0.889	0.851	0.899
	GPC-MV	0.663	0.895	0.692	0.867
	GPC-MA	**0.910**	**0.999**	**0.947**	**0.992**
bupa	GPC-GOLD	1.000	1.000	0.993	1.000
	GPC-CONC	0.862	0.926	0.854	0.932
	GPC-MV	0.793	0.961	0.816	0.953
	GPC-MA	**0.995**	**1.000**	**0.991**	**1.000**
breast	GPC-GOLD	1.000	1.000	0.997	1.000
	GPC-CONC	0.922	0.938	0.936	0.983
	GPC-MV	0.860	0.990	0.887	0.992
	GPC-MA	**0.995**	**1.000**	**0.996**	**1.000**
tic-tac-toe	GPC-GOLD	1.000	1.000	1.000	1.000
	GPC-CONC	0.828	0.887	0.884	0.952
	GPC-MV	0.717	0.932	0.806	0.958
	GPC-MA	**0.999**	**1.000**	**1.000**	**1.000**

Table 1. Average accuracy and AUC over 30 runs, obtained by simulating 7 annotators on different UCI datasets.

Dataset	GOLD	CONC	MV	GPC-MA
ionosphere	0.495	403.618	0.476	2.470
pima	0.551	357.238	0.445	2.583
parkinsons	0.187	55.424	0.186	0.608
bupa	0.551	357.238	0.445	2.583
breast	2.176	3071.467	1.474	8.093
tic-tac-toe	3.67	5035.112	3.106	16.130

Table 2. Average execution times (in seconds) over 30 runs of the different approaches.

In order to compare the different approaches in terms of computational demands, the execution times were also measured. Table 2 shows the average execution times over 30 runs on a Intel Core i7 2600 (3.4GHZ) machine with 32GB DDR3 (1600MHZ) of memory.

The results obtained show that the proposed approach (GPC-MA) consistently outperforms the two baselines in the 6 datasets used, while only raising the computational time by a small factor (between 3 and 5) when compared to the majority voting baseline. Furthermore, we can see that GPC-MA is considerably faster (up to 100x) than the GPC-CONC baseline, which is not surprising since the computational complexity of GPs is $\mathcal{O}(N^3)$ and the dataset used in GPC-CONC is R-times larger than the original dataset. However, GPC-CONC seems to perform better than the other baseline method: GPC-MV. We hypothesize that this is due to the fact that GPC-CONC can model the uncertainty introduced by the heterogeneity in the annotators'

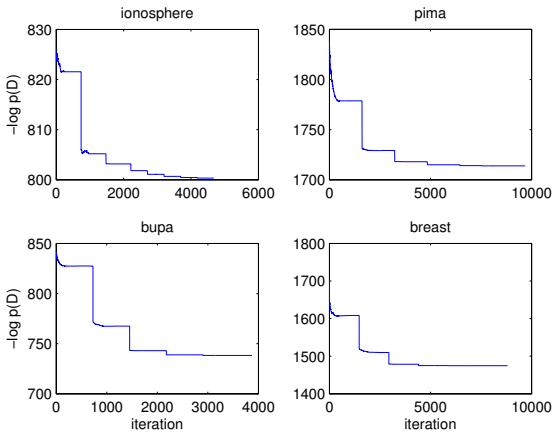

Figure 1. Plots of the log marginal likelihood over 4 runs of GPC-MA using 4 different datasets.

Method	Trainset Accuracy	Trainset AUC	Testset Accuracy	Testset AUC
GPC-GOLD	0.987	0.999	0.723	0.785
GPC-MV	0.886	0.923	0.719	0.781
GPC-MA	**0.900**	**0.944**	**0.721**	**0.783**

Table 3. Results for the sentiment polarity dataset.

Method	Trainset AUC	Trainset F1	Testset AUC	Testset F1
GPC-GOLD	1.000	1.000	0.852	0.683
GPC-CONC	0.926	0.700	0.695	0.423
GPC-MV	0.812	0.653	0.661	0.411
GPC-MA	**0.943**	**0.702**	**0.882**	**0.601**

Table 4. Results obtained for the music genre dataset.

answers. Hence, if for example, all 7 annotators assign the same label to some data point, the variance associated with that data point will be lower than when the 7 annotators provide contradicting labels.

Figure 1 shows plots of the (negative) log marginal likelihood over 4 runs of GPC-MA using 4 different datasets, where it becomes clear the effect of the re-estimation of the annotator's parameters α and β, which is evidenced by the periodic "steps" in the log marginal likelihood.

4.2. Real annotators

The proposed approach was also evaluated on real multiple-annotator settings by applying it to the datasets used in (Rodrigues et al., 2013a) and made available online by the authors. These consist on a sentiment polarity and a music genre classification dataset. The former contains 5000 sentences from movie reviews extracted from the website RottenTomatoes.com and whose sentiment was classified as positive or negative, while the latter contains 700 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. Both datasets were published on Amazon Mechanical Turk for annotation, and the authors collected a total 27747 and 2946 labels for training, corresponding to 203 and 44 distinct annotators, respectively. For both tasks, separate test sets are provided. The test set for the sentiment task consists of 5429 sentences while the test set for the music genre task contains 300 samples. For further details on these datasets, the interested readers are redirected to the original paper (Rodrigues et al., 2013a).

Tables 3 and 4 show the results obtained for the different approaches in the sentiment and music datasets respectively. Since the music dataset corresponds to a multi-class problem, we proceeded by transforming it into 10 different binary classification tasks. Hence, each task corresponds to identifying songs of each genre. Unlike the previous experiments, with the music genre dataset a squared exponential covariance function with Automatic Relevance Determination (ARD) was used, and the hyper-parameters were optimized by maximizing the marginal likelihood.

Due to the computational cost of the GPC-CONC approach and the size of the sentiment dataset, we were unable to test this method on this dataset. Nevertheless, the obtained results show the overall advantage of GPC-MA over the baseline methods.

4.3. Active learning

The active learning heuristics proposed were tested on the music genre dataset from Section 4.2. For each genre, we randomly initialize the algorithm with 200 instances and then perform active learning for another 300 instances. In order to make active learning more efficient, in each iteration we rank the unlabeled instances according to eq. 11 and select the top 10 instances to label. For each of these instances we query the best annotator according to the heuristic we proposed for selecting annotators (eq. 12). Since each instance in the dataset is labeled by an average of 4.21 annotators, picking a single annotator per instance corresponds to savings in annotation cost of more than 76%. Each experiment is repeated 30 times with different random initializations. Figure 2 shows how the average testset AUC for the different music genres evolves as more labels are queried. We compare the proposed active learning methodology with a random baseline. In order to make clear the individual contributions of each of the heuristics proposed, we also show the results of using only the heuristic in eq. 11 for selecting an instance to label and selecting the annotators at random. As the figure evidences, there is a clear advantage in using both active learning heuristics together, which can provide an improvement in AUC of more than 10% after the 300 queries.

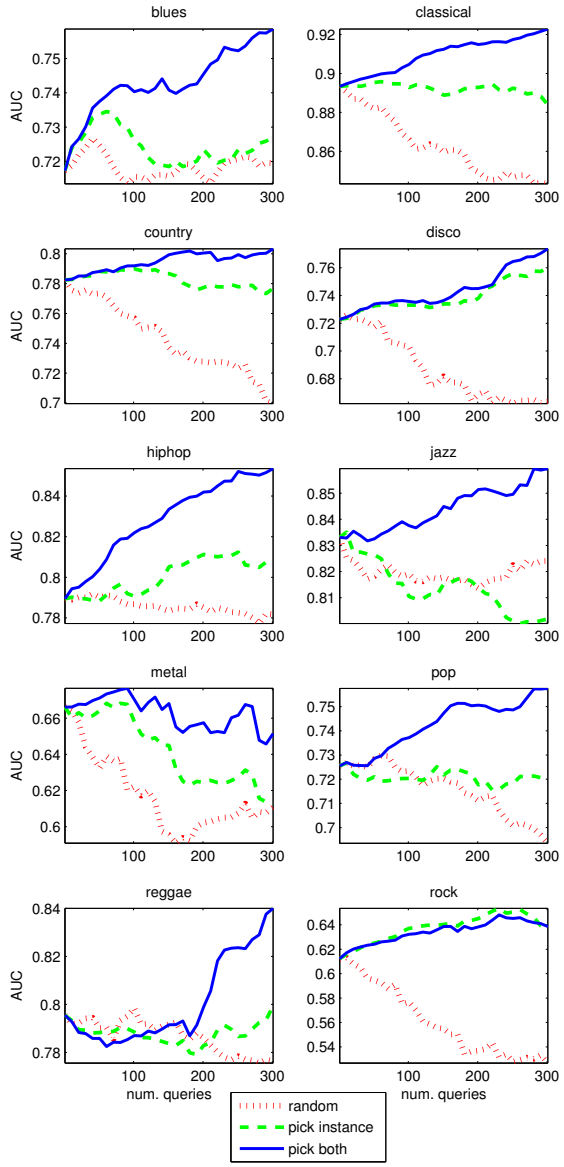

Figure 2. Active learning results on music genre dataset.

5. Conclusion and future work

This paper presented the generalization of the Gaussian process classifier (a special case when $R = 1$, $\alpha = 1$ and $\beta = 1$), a non-linear non-parametric Bayesian classifier, to multiple-annotator settings. By treating the unobserved true labels as latent variables, this model is able to estimate the different levels of expertise of the multiple annotators, thereby being able to compensate for their biases and thus obtaining better estimates of the ground truth labels. We empirically show, using both simulated annotators and real multiple-annotator data collected from Amazon Mechanical Turk, that while this model only incurs in a small increase in the computational cost of approximate Bayesian inference with EP, it is able to significantly outperform all the baseline methods. Furthermore, two simple and yet effective active learning heuristics were proposed, which can provide an even further boost in classification performance, while reducing the number of annotations required, and consequently the annotation cost.

The proposed approach makes the assumption that the labels provided by the different annotators do not depend on the instance their labeling, i.e. $p(y^r|z, \mathbf{x}) = p(y^r|z)$. Future work will try to relax this assumption by considering dependencies on $\mathbf{x}$, and by modeling $p(y^r|z, \mathbf{x})$ with a Gaussian process. Regarding active learning, future work will also explore ways of *jointly* selecting the instance to label and the best annotator to label it.

Acknowledgments

The Fundação para a Ciência e Tecnologia (FCT) is gratefully acknowledged for founding this work with the grants SFRH/BD/78396/2011 and PTDC/EIA-EIA/115014/2009 (CROWDS).

References

Bachrach, Y., Graepel, T., Minka, T., and Guiver, J. How to grade a test without knowing the answers - a Bayesian graphical model for adaptive Crowdsourcing and aptitude testing. In *Proc. of the 29th Int. Conf. on Machine Learning*, 2012.

Barber, D. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Chen, X., Lin, Q., and Zhou, D. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proc. of the 30th Int. Conf. on Machine Learning*, pp. 64–72, 2013.

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

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

Howe, J. *Crowdsourcing: why the power of the Crowd is driving the future of business*. Crown Publishing Group, New York, NY, USA, 1 edition, 2008.

Kapoor, A., Grauman, K., Urtasun, R., and Darrell, T. Active learning with Gaussian processes for object categorization. In *Int. Conf. on Computer Vision (ICCV)*, pp. 1–8, 2007.

Lawrence, N. D., Seeger, M., and Herbrich, R. Fast sparse Gaussian process methods: the informative vector machine. In *Advances in Neural Information Processing Systems 15*, pp. 609–616. MIT Press, 2003.

Liu, C. and Wang, Y. TrueLabel + Confusions: a spectrum of probabilistic models in analyzing multiple ratings. In *Proc. of the 29th Int. Conf. on Machine Learning*, 2012.

Minka, T. Expectation Propagation for approximate Bayesian inference. In *Proc. of the 17th Conference in Uncertainty in Artificial Intelligence*, pp. 362–369, 2001.

Rasmussen, C. E. and Williams, C. *Gaussian processes for machine learning (Adaptive computation and machine learning)*. The MIT Press, 2005.

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

Rodrigues, F., Pereira, F., and Ribeiro, B. Learning from multiple annotators: distinguishing good from random labelers. *Pattern Recognition Letters*, pp. 1428–1436, 2013a.

Rodrigues, F., Pereira, F., and Ribeiro, B. Sequence labeling with multiple annotators. *Machine Learning*, pp. 1–17, 2013b.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. 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, 2008.

Welinder, P., Branson, S., Belongie, S., and Perona, P. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems 23*, pp. 2424–2432, 2010.

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

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

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