Knowledge-Based Systems 261 (2023) 110183

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Probabilistic fusion of crowds and experts for the search of gravitational waves ☆

Pablo Ruiz [a,*], Pablo Morales-Álvarez [b], Scott Coughlin [c], Rafael Molina [d], Aggelos K. Katsaggelos [e]

[a] OriGen.AI, Brooklyn, NY, 11201, US
[b] Department of Statistics and Operations Research, University of Granada, Granada, 18071, Spain
[c] Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA) at Northwestern University, Evanston, IL, 60201, US
[d] Department of Computer Science and AI, University of Granada, Granada, 18071, Spain
[e] Department of Electrical and Computer Engineering at Northwestern University, Evanston, IL, 60208, US

ARTICLE INFO

Article history:
Received 19 June 2021
Received in revised form 3 October 2022
Accepted 4 December 2022
Available online 13 December 2022

Keywords:
Crowdsourcing
Classification
Gravitational waves
(Sparse) Gaussian processes
Variational inference

ABSTRACT

The acquisition of training labels in machine learning classification tasks is expensive. In the last years, crowdsourcing has emerged as a popular approach to label a training set. Crowdsourcing shares the labeling effort among a large number of (possibly non-expert) annotators. Moreover, in many realistic applications, a limited number of expert labels can also be collected to complement the crowdsourcing ones. Such combination of (millions of) crowdsourced and (a few) expert labels is precisely the setting in the GravitySpy project. The goal of GravitySpy is to enhance the detection of gravitational waves, which provide a new way of exploring the early universe in astrophysics (their first detection got the 2017 Physics Nobel prize). In this work, we propose a new probabilistic crowdsourcing model based on sparse Gaussian Processes (GPs) which allows for the integration of expert labels. To the best of our knowledge, this is the first probabilistic GP-based method that tackles this setting. We demonstrate that the resulting objective function to be optimized is a natural fusion of the crowdsourcing and the standard sparse GP classification objectives. Desirable theoretical properties of the crowdsourcing method, translate in a mathematical sound manner into the new method. The new algorithm is implemented in TensorFlow. A controlled experiment illustrates the properties and behavior of the proposed method. We also show that it performs as theoretically expected in a well-known real-world crowdsourcing dataset. Finally, its application to GravitySpy obtains 92.58% overall accuracy and 92.27% test-likelihood, outperforming all previous methods in the literature.

1. Introduction

With classification problems, the amount of training labels has a direct impact on the performance of machine learning algorithms [1]. Consequently, collecting labels for the training data is one of the main steps in real-world problems [2,3]. This step constitutes an important bottleneck in difficult tasks where plenty of expert knowledge is required for labeling, such as medical applications or the classification of complex remote sensing signals [4–6]. In these cases, especially when the dataset is large, only a limited amount of training labels can be provided by experts, and some other labeling scheme must be considered too.

Crowdsourcing, also known as citizen science, has become a popular approach to labeling real-world datasets [7,8]. In the last decade, many crowdsourcing services have proliferated in the internet, where a dataset can be published and millions of people around the world can provide labels in exchange for a reward [9]. Amazon Mechanical Turk (www.mturk.com), Zooniverse (www.zooniverse.org), and Innocentive (www.innocentive.com) are among the most popular ones. Crowdsourcing shares the labeling effort among a large number of annotators with different degrees of expertise. During the last years, many crowdsourcing algorithms have been developed to extract knowledge from the heterogeneous crowdsourcing scenario [10–12].

In particular, the two paradigms (crowdsourcing vs expert labels) should not be regarded as mutually exclusive, and some

☆ This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement No 860627 (CLARIFY Project), the US National Science Foundation through the NSF INSPIRE 15-47880 grant (Gravity Spy project), the Spanish Ministry of Science and Innovation under project PID2019-105142RB-C22, the FEDER/Junta de Andalucía-Consejería de Transformación Económica, Industria, Conocimiento y Universidades under the project P20_00286, and the University of Granada through the Visiting Scholar Program.

* Corresponding author.
E-mail address: mataran@origen.ai (P. Ruiz).

https://doi.org/10.1016/j.knosys.2022.110183
0950-7051/

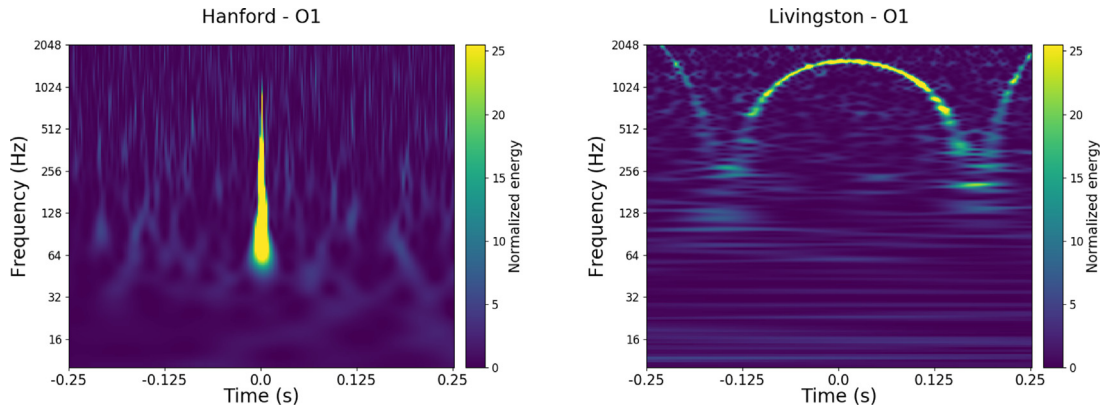

Fig. 1. Two examples of glitches observed by the LIGO detector. The goal of the GravitySpy project is to develop a machine learning system that automatically classifies different types of glitches, in order to improve the detection of gravitational waves. More details about the problem are provided in Section 4.3.

expert labels can be collected to complement the crowdsourcing annotations. In fact, one of the main limitations of crowdsourcing methods, their *identifiability*, may be alleviated by adding some expert labels. The problem of identifiability was already highlighted in the founding work [13, section 3]. Namely, a majority of unreliable annotators would make crowdsourcing methods learn an incorrect concept (indeed, when crowdsourcing methods are used in real practice, it is implicitly assumed that most annotators are reliable). Intuitively, the addition of some expert labels could guide crowdsourcing algorithms to identify the underlying truth, even in cases where there is a majority of unreliable annotators.

Interestingly, the combination of crowdsourcing and expert labels is the setting available within the GravitySpy project [14,15]. GravitySpy aims at classifying glitches produced in the Nobel-laureate Laser Interferometer Gravitational-Waves Observatory (LIGO), see Fig. 1. Whereas the labeling process of GravitySpy has been crowdsourced, astrophysicists have also provided expert labels for a smaller training dataset. The best results with the expert labels alone were obtained by Convolutional Neural Networks [16]. Then, crowdsourcing methods leveraged the larger crowdsourcing dataset to establish a new state-of-the-art solution for this problem [10]. Here, we will show that a probabilistic fusion of both settings outperforms the results achieved separately.

The proposed model, which is named *SVGPCR-Mix*, extends the probabilistic crowdsourcing method *SVGPCR* [10] to integrate expert labels. It is based on (sparse) Gaussian Processes (GPs) [17], which were shown to outperform deep learning crowdsourcing approaches in the GravitySpy data, see [10]. The expertise of annotators to label the different classes is modeled through confusion matrices, which are estimated, along with the rest of model parameters, following a variational inference scheme [18,19]. Interestingly, the derived variational objective (the Evidence Lower Bound, *ELBO*) is a natural fusion of those in [17] (i.e., if sparse GPs were applied on the expert labels alone) and [10] (i.e., if the crowdsourcing labels were used alone).

Synthetic and real data are used to analyze the proposed approach in depth. First, a controlled experiment illustrates the behavior of the method as the amount of expert labels increases. To study the identifiability issues of standard crowdsourcing methods, we simulate different scenarios with different behavior for the annotators. Then, we show that the novel approach also behaves as theoretically expected in the Music Genre dataset, a well-known real-world crowdsourcing dataset. The main difference with the next experiment (the GravitySpy one) is that all expert labels are available in the Music Genre dataset. This allows us to test our method with increasing amounts of expert labels, and compare the results with the gold standard (that is, train a GP classifier with all the true labels). Finally, the proposed probabilistic fusion of crowdsourcing and expert labels is shown to establish a new state-of-the-art approach in the challenging real-world astrophysics application of GravitySpy.

Different properties of the model will be demonstrated with the experiments. Many of them are inherited from *SVGPCR*, such as the scalability to large datasets, the estimation of the annotators confusion matrices (i.e., their degree of expertise), and the estimation of the ground truth for the crowdsourced samples. Some others are specific to *SVGPCR-Mix*, such as addressing the identifiability issues of crowdsourcing methods, the role of anchor points for the expert labels, and the relevance of the coupling term in the ELBO.

The main contributions of this work are summarized on the following list:

- We extend the probabilistic model in *SVGPCR* [10] to address the problem of jointly training with expert and crowdsourced labels. To the best of our knowledge, this is the first probabilistic GP-based machine learning algorithm that allows for fusing knowledge obtained from crowds and experts.
- We show that the derived objective function is a natural fusion of the crowdsourcing [10] and the standard sparse GP classification [17] objectives.
- We demonstrate that the proposed model leverages the expert labels to solve the identifiability issues of methods trained with crowdsourcing labels only, see the synthetic experiment.
- The proposed model is tested on two real-world problems, outperforming state-of-the-art methods.

This paper is organized as follows. Section 2 is dedicated to related works. The proposed model and inference are presented in Sections 3.1 and 3.2, respectively. Sections 4.1, 4.2 and 4.3 contain the controlled experiment, the study on the Music Genre dataset, and the application to LIGO data, respectively. Section 5 concludes the paper.

2. Related work

The proposed method is the first probabilistic model based on GPs that utilizes expert and crowdsourcing labels in conjunction. In this section we first review the literature on the GP concepts (non-crowdsourcing labels) needed in this work, and then, examine the state of the art for crowdsourcing methods paying special attention to probabilistic methods.

First GP formulations have been proposed for regression and standard classification problems [20,21]. The main limitation of

these models was their scalability. They could not be applied on large datasets because they required the inversion of a kernel matrix (of size $N \times N$, with N the size of the training set) in each iteration of the training algorithm. To mitigate this problem, Snelson and Ghahramani proposed the sparse GP [22], which introduced the concept of inducing points, which refers to a smaller set of M ($M \ll N$) samples that condense the information contained in the training set. Later, Hensman et al. [17,23] introduced a new method using Variational Inference and inducing points, first for regression in [23] and then for classification in [17]. As we will discuss in Section 3.2, our method is a generalization of the one in [17], and coincides with it when only expert labels are available.

Regarding crowdsourcing methods, it is widely accepted that the first paper to address crowdsourcing problems was [13] published in 1979. Initial approaches to deal with crowdsourcing labels relied on label aggregation mechanisms prior to training. The most simplest example is majority voting, which assumes that every annotator is equally reliable. More elaborated methods such as [24,25] consider the biases of the different annotators, yielding a better calibrated set of training labels. These initial methods worked only with the labels provided by the annotators and they did not take into account observed features, which means that these models unrealistically assumed that the difficulty to label a sample was always the same. To avoid that problem, Raykar et al. [26,27] introduced a two-class method based on logistic regression that took into account observed features. The annotators' behavior is modeled with sensitivity and specificity values which were estimated during training. Raykar's works can be considered the cornerstones on which most of the subsequent probabilistic modeling and inference works on crowdsourcing are based. However, they had an important drawback, the underlying logistic regression classifier did not allow to learn complex classification functions. To solve that problem, Rodrigues et al. [28] introduced the first GP model trained with crowdsourcing labels, which used Expectation-Maximization as inference method [29]. Later, Ruiz et al. [19] proposed the use of Variational Inference. However, these two methods suffer from the scalability problem inherited from GP, recall the review of GPs at the beginning of this section. Two different approaches were proposed by Morales-Álvarez et al. [10,30] to address it. First, in [30] the authors used Random Fourier Features which can be understood as a scalable approximation of a RBF kernel. In the second method, Morales-Álvarez et al. [10] introduced *SVGPCR*. This method was also recently used by López-Pérez et al. [8] to detect breast cancer in histology images. *SVGPCR* is a multi-class method where annotators' behavior is modeled with confusion matrices, and the scalability problem is solved using inducing points. As we will discuss in Section 3.2, our method also generalizes [10], and coincides with it when only crowdsourcing labels are available.

In addition to probabilistic methods, there are several Deep Learning methods that had an important impact in crowdsourcing literature. First, Albarquoni et al. [31] introduced *Aggnet*, the first neural network trained with crowdsourcing labels. It was applied to a two-class classification problem, mitosis detection in breast cancer histology images. Later, Rodrigues et al. [32] proposed a more general method that also addressed multi-class problems. They introduced the *crowd layer* which can be added at the end of any classification neural network in order to train it with crowdsourcing labels.

It is important to note that all these crowdsourcing methods share the identifiability problem, that is, they will learn wrong patterns in scenarios with majority of unreliable annotators (e.g. adversarial or spammer annotators, which will be described in Section 4.1). More importantly, the integration of expert labels provides our model with valuable information to learn what the real underlying truth is, and thus detects annotators who are not reliable.

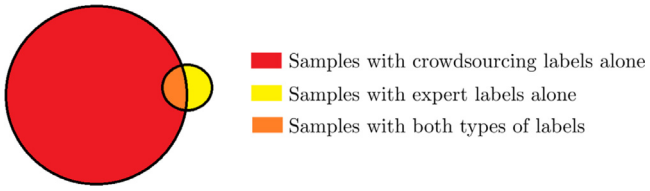

Fig. 2. Distribution of expert and crowdsourcing labels across the training samples. There may be samples with only crowdsourcing labels, others with only expert labels, and others with both.

3. Probabilistic modeling and inference

In this section we introduce the proposed method. Namely, the probabilistic modeling is explained in Section 3.1. Then, variational inference is detailed in Section 3.2, where we also explain how to make predictions on test instances.

3.1. Probabilistic model

In this section, we describe the proposed model. Before going into the mathematical details, let us sketch the intuition behind it. The only available data in the training step is given by the features, the crowdsourcing labels, and a few expert labels. In order to jointly model these three components, we assume that each instance has an underlying true/correct label. A few of them are known (i.e. the expert labels), but most of them are unknown. The crowdsourcing labels given by each annotator are modeled based on these true labels and a confusion matrix associated to each annotator, which is estimated too. Therefore the proposed model also estimates the degree of expertise for each annotator (given by his/her confusion matrix). In the rest of this section we introduce the notation and the full details for the probabilistic model. Fig. 3 shows the graphical representation of the proposed model, which will be helpful throughout this section.

Notation. Let $\mathbf{X} \in \mathbb{R}^{N \times D}$ be an (observed) training set of N D-dimensional samples. The (mostly un-observed) true labels are denoted as $\mathbf{Z} \in \{0, 1\}^{N \times K}$, where each one of the K classes is expressed through an one-hot encoding. Two types of information are available for training: i) a few expert (i.e., true) labels, and (ii) crowdsourcing annotations. As shown in Fig. 2, there may be samples with crowdsourcing annotations alone, other samples with expert labels alone, and other samples for which both types of information are available.

Regarding the expert labels, let $\mathcal{O} \subseteq \{1, \dots, N\}$ denote the samples for which the expert label is observed (and let $\mathcal{U} = \{1, \dots, N\} \setminus \mathcal{O}$ refer to the rest). That is, looking at Fig. 2, $\mathcal{O}$ refers to the yellow and orange regions, and $\mathcal{U}$ refers to the red one. We split $\mathbf{Z}$ into $\mathbf{Z}_{\mathcal{O}}$ (the observed true labels) and $\mathbf{Z}_{\mathcal{U}}$ (the un-observed, most of them). With regards to the crowdsourcing annotations, let A be the number of annotators, $\mathcal{A}_n \subseteq \{1, \dots, A\}$ the subset of annotators who labeled the nth sample (this is empty for the samples in the yellow region), and $\mathbf{Y}_n^a$ the set of labels provided by the ath annotator for that sample.[1] All crowdsourcing labels (for all samples and annotators) are jointly denoted as $\mathbf{Y}$.

Modeling the crowdsourcing annotations given the true labels. The behavior of each annotator a ($a = 1, \dots, A$) is modeled using a $K \times K$ confusion matrix $\mathbf{R}^a = (r_{ij}^a)$, $1 \leq i, j \leq K$. Specifically, r_{ij}^a is the probability that annotator a provides the label i for a sample whose real class is j. This is mathematically

[1] Although $\mathbf{Y}_n^a$ typically contains only one label, it is straightforward to model the case when an annotator provides more than one label for the same sample, which happens in the GravitySpy data.

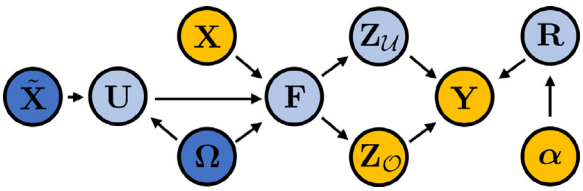

Fig. 3. Probabilistic graphical model for *SVGPCR-Mix*. Yellow nodes are observed, light blue nodes are inferred through a posterior distribution, and dark blue nodes are inferred with point estimates.

expressed as $\mathrm{p}(\mathbf{y}|\mathbf{z}, \mathbf{R}^a) = \mathbf{y}^\intercal \mathbf{R}^a \mathbf{z}$, where $\mathbf{y} \in \mathbf{Y}$ and $\mathbf{z} \in \mathbf{Z}$. Assuming that annotators label samples independently, the full observation model for the crowdsourcing labels is the following product

$$\mathrm{p}(\mathbf{Y}|\mathbf{Z}, \mathbf{R}) = \prod_{n=1}^{N} \prod_{a \in \mathcal{A}_n} \prod_{\mathbf{y} \in \mathbf{Y}_n^a} \mathrm{p}(\mathbf{y}|\mathbf{z}_n, \mathbf{R}^a). \tag{1}$$

Prior knowledge about the annotators' behavior is modeled with a (conjugate) Dirichlet distribution:

$$\mathrm{p}(\mathbf{R}) = \prod_{a=1}^{A} \prod_{k=1}^{K} \mathrm{p}(\mathbf{r}_k^a) = \prod_{a=1}^{A} \prod_{k=1}^{K} \mathrm{Dir}(\mathbf{r}_k^a | \alpha_{1k}^a, \ldots, \alpha_{Kk}^a), \tag{2}$$

where $\mathbf{r}_k^a = (r_{1k}^a, \ldots, r_{Kk}^a)^\intercal$ is the kth column of $\mathbf{R}^a$ and $\boldsymbol{\alpha} = \{\alpha_{ij}^a : i, j = 1, \ldots, K,\ a = 1, \ldots, A\}$ are prior hyperparameters. If there is no prior information on annotator a, we set $\alpha_{ij}^a = 1$ for all $i, j = 1, \ldots, K$, which produces a uniform prior. Notice that, even when there is no prior information about the annotators, the use of a prior distribution protects us from the so-called "black swan paradox" [33, Section 3.3.4.1]. Namely, if the annotator a did not provide any labels for samples in class j, then there would be no information to infer the column $\mathbf{r}_j^a$ of $\mathbf{R}^a$.

Modeling the true labels with GPs. To relate the true labels $\mathbf{Z}$ and the observed features $\mathbf{X}$, we resort to GPs [21], which have proven successful in crowdsourcing [10,28]. We introduce latent variables $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_K] \in \mathbb{R}^{N \times K}$ (one vector per class), for which the following GP prior is considered:

$$\mathrm{p}(\mathbf{F}|\mathbf{X}, \boldsymbol{\Omega}) = \prod_{k=1}^{K} \mathrm{p}(\mathbf{f}_k|\mathbf{X}, \boldsymbol{\Omega}_k) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{f}_k|\mathbf{0}, \mathbf{K}_{\boldsymbol{\omega}_k}(\mathbf{X})). \tag{3}$$

A standard Radial Basis Function (RBF) kernel, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \gamma \cdot \exp\{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2)\}$, is used for the GPs, whose parameters $\boldsymbol{\Omega} = \{\boldsymbol{\omega}_k\}_{k=1}^K = \{\gamma_k, \sigma_k^2\}_{k=1}^K$ are estimated during training (inference is discussed in the next section).

The relationship between the true labels $\mathbf{Z}$ and the latent variables $\mathbf{F}$ is modeled through the Robust-Max likelihood [34]. Specifically, assuming independence between the true labels given the latent variables, we have

$$\mathrm{p}(\mathbf{Z}|\mathbf{F}) = \prod_{n=1}^{N} \mathrm{p}(\mathbf{z}_n|\mathbf{f}_{n,:}) = \mathrm{p}(\mathbf{Z}_{\mathcal{O}}|\mathbf{F})\mathrm{p}(\mathbf{Z}_{\mathcal{U}}|\mathbf{F}), \tag{4}$$

where we have explicitly split $\mathbf{Z}$ into $\mathbf{Z}_{\mathcal{O}}$ and $\mathbf{Z}_{\mathcal{U}}$. Notice that both terms are conceptually different: while $\mathbf{Z}_{\mathcal{O}}$ is observed (along with $\mathbf{Y}$), $\mathbf{Z}_{\mathcal{U}}$ is unknown and will be estimated within the variational inference scheme (next section).

Addressing GPs scalability issues. One of the main limitations of GPs is their scalability [21,30,35]. Their training cost is $\mathcal{O}(N^3)$, which hampers their application beyond a few thousand samples (typically 10K). Since the GravitySpy set is much larger than this, we sparsify our GP based on standard inducing points approaches [17]. Namely, latent variables $\mathbf{F}$ are extended with $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K] \in \mathbb{R}^{M \times K}$, where $M \ll N$. These variables are called inducing points, and represent the values of the GP at M different inducing locations $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_M]^\intercal \in \mathbb{R}^{M \times D}$.

Summary of the proposed joint model. In conclusion, the full model, including the inducing points, is given by the product of all the distribution defined so far:

$$\begin{aligned}\mathrm{p}(\mathbf{Y}, \mathbf{Z}_{\mathcal{O}}, \mathbf{Z}_{\mathcal{U}}, \mathbf{F}, \mathbf{U}, \mathbf{R}|\boldsymbol{\Omega}, \tilde{\mathbf{X}}) = \mathrm{p}(\mathbf{Y}|\mathbf{Z}, \mathbf{R})\mathrm{p}(\mathbf{Z}_{\mathcal{O}}|\mathbf{F})\mathrm{p}(\mathbf{Z}_{\mathcal{U}}|\mathbf{F})\mathrm{p}(\mathbf{F}|\mathbf{U}, \boldsymbol{\Omega}) \\ \times \mathrm{p}(\mathbf{U}|\boldsymbol{\Omega})\mathrm{p}(\mathbf{R}).\end{aligned} \tag{5}$$

Notice that, in order to lighten the notation, we have omitted the dependency of $\mathrm{p}(\mathbf{F}|\mathbf{U}, \boldsymbol{\Omega})$ and $\mathrm{p}(\mathbf{U}|\boldsymbol{\Omega})$ on $\tilde{\mathbf{X}}$ (just like $\mathbf{X}$ is not explicitly shown).

Knowledge modeling. The described model offers different alternatives to model previous knowledge that may be available depending on the application. The expert labels $\mathbf{Z}_{\mathcal{O}}$ provide the most straightforward way to introduce knowledge on the true labels of instances. Another potential source of knowledge is the behavior of annotators. Such knowledge can be modeled through the prior distribution $\mathrm{p}(\mathbf{R})$. For instance, if we know that annotator a tends to classify class j as class i, we can codify this by increasing the value of α_{ij}^a in the Dirichlet prior. Likewise, the smoothness of the underlying function can be controlled by the type of kernel used for the GP prior.

3.2. Variational inference

The goal in this section is to infer the unknown variables in the model, i.e., the blue ones in Fig. 3. Specifically, we want to calculate the posterior distribution over the variables $\boldsymbol{\Phi} = \{\mathbf{Z}_{\mathcal{U}}, \mathbf{F}, \mathbf{U}, \mathbf{R}\}$, and obtain point estimates for the kernel hyperparameters $\boldsymbol{\Omega}$ and the inducing point locations $\tilde{\mathbf{X}}$. In principle, this requires the integration of the joint distribution in Eq. (5) with respect to all the variables $\boldsymbol{\Phi}$. Since this integral is analytically intractable, we resort to variational inference (VI) [18], which casts inference as an optimization problem.

Leveraging VI for our model. Specifically, the log likelihood of the model can be decomposed as follows

$$\begin{aligned}\log \mathrm{p}(\mathbf{Y}, \mathbf{Z}_{\mathcal{O}}|\boldsymbol{\Omega}, \tilde{\mathbf{X}}) =& \mathrm{KL}(\mathrm{q}(\boldsymbol{\Phi}) \| \mathrm{p}(\boldsymbol{\Phi}|\mathbf{Y}, \mathbf{Z}_{\mathcal{O}}, \boldsymbol{\Omega}, \tilde{\mathbf{X}})) \\ &+ \underbrace{\int \mathrm{q}(\boldsymbol{\Phi}) \log \frac{\mathrm{p}(\mathbf{Y}, \mathbf{Z}_{\mathcal{O}}, \boldsymbol{\Phi}|\boldsymbol{\Omega}, \tilde{\mathbf{X}})}{\mathrm{q}(\boldsymbol{\Phi})} \mathrm{d}\boldsymbol{\Phi}}_{\text{ELBO}},\end{aligned} \tag{6}$$

which is valid for any probability distribution $\mathrm{q}(\cdot)$ of the unknown variables $\boldsymbol{\Phi}$. The right-hand side in Eq. (6) is the sum of two terms: the Kullback–Leibler (KL) divergence term and the Evidence Lower Bound (ELBO). The KL divergence is always non-negative, and it is equal to zero if and only if $\mathrm{q}(\boldsymbol{\Phi})$ coincides with the sought true posterior distribution $\mathrm{p}(\boldsymbol{\Phi}|\mathbf{Y}, \mathbf{Z}_{\mathcal{O}}, \boldsymbol{\Omega}, \tilde{\mathbf{X}})$. Therefore, the optimal posterior distribution over $\boldsymbol{\Phi}$ can be obtained by minimizing the KL term. Regarding the parameters $\boldsymbol{\Omega}$ and $\tilde{\mathbf{X}}$, they must be optimized to maximize the log likelihood of the model, i.e., left-hand side of Eq. (6). Interestingly, both tasks (minimizing the KL divergence on $\mathrm{q}(\boldsymbol{\Phi})$ and maximizing the log likelihood of the model on $\boldsymbol{\Omega}$ and $\tilde{\mathbf{X}}$) can be jointly accomplished by maximizing the ELBO with respect to $\mathrm{q}(\boldsymbol{\Phi})$, $\boldsymbol{\Omega}$ and $\tilde{\mathbf{X}}$. This is indeed the training objective for VI.

The proposed parametric posterior distribution. To optimize with respect to the probability distribution $\mathrm{q}(\boldsymbol{\Phi})$, VI assumes a parametric form $\mathrm{q}_{\boldsymbol{\Theta}}(\boldsymbol{\Phi})$ for it, and optimizes with respect to these parameters $\boldsymbol{\Theta}$, which are called the variational parameters. In this work we assume that $\mathrm{q}(\boldsymbol{\Phi})$ factorizes as:

$$\mathrm{q}(\mathbf{Z}_{\mathcal{U}}, \mathbf{F}, \mathbf{U}, \mathbf{R}) = \mathrm{q}(\mathbf{Z}_{\mathcal{U}})\mathrm{q}(\mathbf{F}|\mathbf{U}, \boldsymbol{\Omega})\mathrm{q}(\mathbf{U})\mathrm{q}(\mathbf{R}), \tag{7}$$

with

$$\mathrm{q}(\mathbf{Z}_{\mathcal{U}}) = \prod_{n \in \mathcal{U}} \mathrm{q}(\mathbf{z}_n) = \prod_{n \in \mathcal{U}} \mathbf{z}_n^\intercal \mathbf{q}_n, \tag{8}$$

$$q(\mathbf{F}|\mathbf{U},\Omega) = p(\mathbf{F}|\mathbf{U},\Omega), \tag{9}$$

$$q(\mathbf{U}) = \prod_{k=1}^{K} q(\mathbf{u}_k) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{u}_k|\mathbf{m}_k,\mathbf{S}_k), \tag{10}$$

$$q(\mathbf{R}) = \prod_{a=1}^{A}\prod_{k=1}^{K} q(\mathbf{r}_k^a) = \prod_{a=1}^{A}\prod_{k=1}^{K} \mathrm{Dir}(\mathbf{r}_k^a|\tilde{\alpha}_{1k}^a,\ldots,\tilde{\alpha}_{Kk}^a). \tag{11}$$

The variational parameters of this posterior, which are jointly denoted as Θ, are:

1. the ground truth estimation for $\mathbf{Z}_{\mathcal{U}}$, i.e., $\mathbf{q}_n = (q_{n1},\ldots,q_{nK})$, $q_{nk} \geq 0$, $\sum_k q_{nk} = 1$, $n \in \mathcal{U}$;
2. the means and covariances in the inducing points, i.e., $\{\mathbf{m}_k, \mathbf{S}_k : k = 1,\ldots,K\}$;
3. the posterior Dirichlet parameters, i.e., $\{\tilde{\alpha}_{ij}^a > 0,\ \ i,j = 1,\ldots,K,\ \ a = 1,\ldots,A\}$.

Deriving and understanding the expression for the ELBO. With this parametric form for the approximate posterior, the ELBO term is given by the following expression, where $\mathbf{e}_k$ denotes the kth K-dimensional one-hot encoding vector:

$$\begin{aligned}
\mathrm{ELBO}(\Omega,\tilde{\mathbf{X}},\Theta) &= \sum_{n\in\mathcal{U}}\sum_{a\in\mathcal{A}_n}\sum_{\mathbf{y}\in\mathbf{Y}_n^a}\sum_{k=1}^{K} q_{nk}\mathbb{E}_{q(\mathbf{r}_k^a)}\log p(\mathbf{y}|\mathbf{e}_k,\mathbf{r}_k^a)\\
&+ \sum_{n\in\mathcal{U}}\sum_{k=1}^{K} q_{nk}\mathbb{E}_{q(\mathbf{f}_{n,:})}\log p(\mathbf{e}_k|\mathbf{f}_{n,:}) - \sum_{a=1}^{A}\sum_{k=1}^{K}\mathrm{KL}(q(\mathbf{r}_k^a)\|p(\mathbf{r}_k^a))\\
&- \sum_{n\in\mathcal{U}}\sum_{k=1}^{K} q_{nk}\log q_{nk} - \sum_{k=1}^{K}\mathrm{KL}(q(\mathbf{u}_k)\|p(\mathbf{u}_k))\\
&+ \sum_{n\in\mathcal{O}}\mathbb{E}_{q(\mathbf{f}_{n,:})}\log p(\mathbf{z}_n|\mathbf{f}_{n,:}) + \sum_{n\in\mathcal{O}}\sum_{a\in\mathcal{A}_n}\sum_{\mathbf{y}\in\mathbf{Y}_n^a}\mathbb{E}_{q(\mathbf{r}_{\mathbf{z}_n}^a)}\log p(\mathbf{y}|\mathbf{z}_n,\mathbf{r}_{\mathbf{z}_n}^a).
\end{aligned} \tag{12}$$

Training consists in maximizing this objective function w.r.t. the variational parameters Θ, the kernel hyperparameters Ω and the inducing point locations $\tilde{\mathbf{X}}$. As optimizer we use Adam with default settings [36].

Interestingly, the first five terms of the ELBO are those obtained in *SVGPCR* [10], i.e., when only crowdsourcing labels are available. The fifth and sixth terms coincide with the ELBO of *SVGP* [17], i.e., when a sparse GP is used only on the true labels. The seventh term does not appear in any of the objective functions of [10] or [17]. Notice that $\mathbf{y}$ and $\mathbf{z}_n$ are observed and the only unknown to be estimated is $\mathbf{r}_{\mathbf{z}_n}^a$. In other words, this term couples both parts in the presence of samples that have both expert and crowdsourcing labels (that is, the orange region in Fig. 2). Thus, it contributes to learning the behavior of annotators by comparing both types of labels, and its role will be analyzed in the experiments (Fig. 8). In other words, the proposed method is a natural generalization of both *SVGP* and *SVGPCR*.

As in the case of *SVGPCR* and *SVGP*, the ELBO in Eq. (12) allows for training in mini-batches (the seventh term also factorizes across data points). The computational cost is the same as in *SVGPCR*, i.e., $\mathcal{O}(N_b(M^2 + A_b K))$, where N_b is the number of samples in the minibatch and A_b is the average number of annotations per instance (in the minibatch).

Summary of the training process. The full training procedure is summarized in Algorithm 1. Notice that it is similar to the training process for *SVGP* [17] and *SVGPCR* [10]. In particular, notice that the GP kernel hyperparameters are optimized during training to maximize the ELBO. Regarding the initializations mentioned in Algorithm 1, the kernel hyperparameters Ω and the inducing points locations $\tilde{\mathbf{X}}$ are initialized by training a standard *SVGP* on the available true labels. As for the variational parameters (denoted jointly as Θ), $\mathbf{q}_n$, $\mathbf{m}_k$ and $\mathbf{S}_k$ are also initialized with the same *SVGP*. Finally, $\tilde{\alpha}_{ij}^a$ is initialized using the crowdsourcing annotations given by each annotator and the probabilities for each class obtained in $\mathbf{q}_n$. The algorithm is implemented using TensorFlow and GPflow [37], which leverage automatic differentiation for computing gradients (this is specially useful for Eq. (12)). To ensure reproducibility and extensibility, the code is publicly available at https://ccia.ugr.es/vip/resources/SVGPCRMix.html.

Algorithm 1 Training procedure for *SVGPCR-Mix*.

Input : Training data $\mathbf{X}$, crowdsourcing labels $\mathbf{Y}$, and observed expert labels $\mathbf{Z}_{\mathcal{O}}$.
Initialize variational parameters Θ, GP kernel hyperparameters Ω, inducing point locations $\tilde{\mathbf{X}}$.
foreach *batch of samples* $B \subset [1,\ldots,N]$ **do**
  Consider $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}_{\mathcal{O}}$ restricted to the corresponding batch, i.e. $\mathbf{X}_B$, $\mathbf{Y}_B$ and $(\mathbf{Z}_{\mathcal{O}})_B$.
  Calculate ELBO($\Theta,\Omega,\tilde{\mathbf{X}}$) for the corresponding batch using eq. (12).
  Gradient step w.r.t. Θ, Ω and $\tilde{\mathbf{X}}$ using Adam optimizer with default parameters.
Output: Variational parameters Θ, GP kernel hyperparameters Ω, inducing point locations $\tilde{\mathbf{X}}$.

Understanding the estimated distributions and how to make predictions on test instances. Once the ELBO is maximized, the estimated values for Ω, $\tilde{\mathbf{X}}$ and Θ are substituted into Eq. (7) to fully determine the approximate posterior $q(\mathbf{Z}_{\mathcal{U}},\mathbf{F},\mathbf{U},\mathbf{R})$. This distribution summarizes all the information extracted from the observed data $\{\mathbf{Y},\mathbf{Z}_{\mathcal{O}},\mathbf{X}\}$. Interestingly, each factor into which $q(\cdot)$ is decomposed (see Eq. (7)) has a different purpose. Firstly, the choice of $q(\mathbf{F}|\mathbf{U},\Omega)$ being equal to the prior conditional, recall Eq. (9), allows for the cancellation of both terms in Eq. (6). This is crucial for training in mini-batches, and therefore for the scalability of the proposed method (see [17,38] for more details on sparse GPs). Secondly, $q(\mathbf{Z}_{\mathcal{U}})$ contains the estimated ground truth for the training samples which do not have true labels. Thirdly, the estimated behavior for the annotators is encoded in $q(\mathbf{R})$. Finally, $q(\mathbf{U})$ allows for predicting on new samples $\mathbf{x}^*$ by conditioning on the inducing points [10,17]. More specifically, given a previously unseen test sample $\mathbf{x}^*$, the distribution for its latent variables $\mathbf{f}^*$ is calculated as

$$\begin{aligned}
p(f_k^*|\mathbf{x}^*) &= \int p(f_k^*|\mathbf{x}^*,\mathbf{u}_k)p(\mathbf{u}_k|\mathbf{Y},\mathbf{Z}_{\mathcal{O}})d\mathbf{u}_k \approx \mathbb{E}_{q(\mathbf{u}_k)}p(f_k^*|\mathbf{u}_k)\\
&= \mathcal{N}\left(f_k^*|\mathbf{B}_{\mathbf{x}^*\tilde{\mathbf{X}}}\mathbf{m}_k,\ k_{\mathbf{x}^*\mathbf{x}^*} + \mathbf{B}_{\mathbf{x}^*\tilde{\mathbf{X}}}\left(\mathbf{S}_k - \mathbf{K}_{\tilde{\mathbf{X}}\tilde{\mathbf{X}}}\right)\mathbf{B}_{\tilde{\mathbf{X}}\mathbf{x}^*}\right),
\end{aligned} \tag{13}$$

where $\mathbf{B}_{\mathbf{x}^*\tilde{\mathbf{X}}}$ stands for $\mathbf{K}_{\mathbf{x}^*\tilde{\mathbf{X}}}\mathbf{K}_{\tilde{\mathbf{X}}\tilde{\mathbf{X}}}^{-1}$. The predictive distribution for the label $\mathbf{z}^*$ is then obtained as $p(\mathbf{z}^*) = \int p(\mathbf{z}^*|\mathbf{f}^*)p(\mathbf{f}^*)d\mathbf{f}^*$, which can be computed with numerical methods, e.g., Monte Carlo sampling. The probabilistic graphical model for making predictions on new samples is shown in Fig. 4.

4. Experimental results

The experimental evaluation is organized as follows. Section 4.1 includes a controlled experiment to illustrate the behavior and properties of the proposed method. Section 4.2 shows that it also performs as theoretically expected in a well-known real-world crowdsourcing dataset for which all expert labels are available. Finally, Section 4.3 shows that our approach establishes a new state-of-the-art method in a challenging real-world astrophysics problem: glitch classification in signals acquired by the Laser Interferometer Gravitational-wave Observatory (LIGO).

4.1. Controlled experiment

Problem formulation. A two-class synthetic dataset is considered on $(-\pi,\pi)$. For each $x \in (-\pi,\pi)$, its class is given by the

Table 1
Sensitivity and specificity values in the three different scenarios: majority of *good*, *adversarial* or *spammer* annotators.

	Annotator ID	Sensitivities					Specificities				
		1	2	3	4	5	1	2	3	4	5
Majority of * annotators	*good*	0.9	0.7	0.8	0.9	0.1	0.6	0.8	0.5	0.8	0.2
	adversarial	0.1	0.3	0.2	0.1	0.9	0.4	0.2	0.5	0.2	0.8
	spammer	0.5	0.57	0.48	0.49	0.9	0.45	0.51	0.5	0.51	0.8

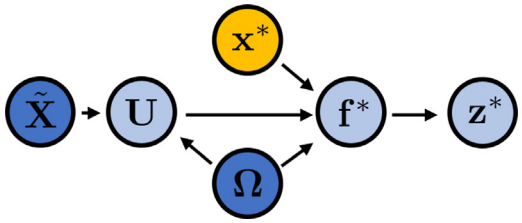

Fig. 4. Probabilistic graphical model for *SVGPCR-Mix* predictions. Once the proposed model is trained, we have the values for $\tilde{\mathbf{X}}$ and $\boldsymbol{\Omega}$ (point estimates), and $\mathbf{U}$ (distribution q($\mathbf{U}$)). Then, given a new sample $\mathbf{x}^*$, the distributions of $\mathbf{f}^*$ and $\mathbf{z}^*$ can be computed to predict its label, see Eq. (13). Yellow nodes are observed, light blue nodes are inferred through a posterior distribution, and dark blue nodes are inferred using point estimates.

Table 2
Training and testing times for the four compared methods (in seconds). We show the training time when using an increasing percentage of expert labels. The reported results are the mean over ten independent runs. Regarding the missing values in the table, notice that, by definition, *GPFull* is only trained with 100% of expert labels, *GPSubset* cannot be trained when there is 0% expert labels, and *SVGPCR* does not use expert labels.

% of expert labels	Training time						Testing time
	0%	20%	40%	60%	80%	100%	
GPFull	–	–	–	–	–	4.12	0.32
GPSubset	–	1.48	1.59	1.99	2.81	4.47	0.34
SVGPCR	4.24	–	–	–	–	–	0.34
SVGPCR-Mix	4.24	5.64	4.42	5.01	5.64	7.63	0.38

sign of cos(3x), i.e., $x \in \mathcal{C}_1$ if cos(3x) > 0 and $x \in \mathcal{C}_0$ otherwise. The ground truth can be seen in Fig. 6 (GT curve). Notice that the classes are not linearly separable.

Crowdsourcing annotations are simulated on 100 randomly distributed samples on $(-\pi, \pi)$. Specifically, to analyze identifiability issues, three different scenarios are considered with a majority of *good*, *adversarial*, or *spammers* annotators. In each scenario, five annotators are simulated. Annotators are modeled by their sensitivity and specificity (i.e., the entries r_{11} and r_{00} of their confusion matrix, respectively). Specifically, the values of sensitivity and specificity for the *good* annotators are high (i.e., they learned the correct concept), for the *adversarial* annotators are low (i.e., they learned the wrong concept), and for the *spammers* are around 0.5 (i.e., they provide a random label). The exact values for each annotator in each scenario are reported in Table 1. Notice that each annotator labels all 100 samples.

***SVGPCR-Mix* outperforms related approaches and alleviates the identifiability issue.** First, we illustrate the performance of *SVGPCR-Mix* as the amount of expert labels grows from 2% (so that there is at least one sample of each class) to 100%.[2] It is compared to three closely related methods. The first two separately rely on the two sources of available information: *SVGPCR* (if only the crowdsourcing labels were available), and *GPSubset* (if a GP was applied on the true labels only). The third, *GPFull*, represents the ideal case when expert labels are available for all the training points and a GP is trained on them (this must be understood as a golden reference). Results are shown in Fig. 5, for the three different scenarios, and averaged over 10 independent runs (a test set of size 1000 is used).

Several aspects can be highlighted in this figure. In the first place, the *SVGPCR-Mix* performance improves with the amount of expert labels, approaching the golden reference *GPFull*. Moreover, the curves saturate quickly (earlier than 20%), which supports the idea that just a few expert labels are needed to complement the crowdsourcing ones. In second place, notice that *SVGPCR* exhibits difficulties when annotators become less reliable (due to the identifiability issues of crowdsourcing methods). Interestingly, *SVGPCR-Mix* requires just a small percentage of expert labels to fix this. In third place, as theoretically expected, *SVGPCR-Mix* stays above *SVGPCR* and *GPSubset* (which uses only one of the sources of information).

Assessing training and testing times. This first experiment is complemented with Table 2, which reports the average training and testing times for each method. All the compared methods are very similar in terms of testing time (production time), since all of them rely on GP prediction. As for the training time, all the results are in the same order of magnitude, although, as expected, times tend to grow when increasing the percentage of expert labels. Likewise, the training time for *SVGPCR-Mix* (which uses both crowdsourcing and expert labels) is greater than that for the rest of methods (which use only one source of labels).

Analyzing the role expert labels as anchor points. The second experiment analyzes how the expert labels behave as anchor points to improve the performance of *SVGPCR-Mix*. Fig. 6 shows the predictive distribution of the model as the amount of expert labels grows (a majority of *spammers* scenario is considered). If crowdsourcing labels are used only (0% curve), the information is so noisy that kernel hyperparameters converge to zero and the predictive distribution is constant (recall we are in the majority-of-*spammers* scenario, where the identifiability limitation of crowdsourcing methods is stronger). The first significant change happens when 5% of expert labels are added (these labels are depicted as orange dots). The accuracy (threshold=0.5) is close to the one provided by the ground truth (GT) model, but the actual posterior probability values are not. The second change occurs with 11% of expert labels (additional labels are depicted as blue crosses). Now, the predictive distribution approximates the ground truth very accurately. Interestingly, notice that no true labels were added in the connected component containing -1; however, the model learned the connection between true and crowdsourcing labels in other regions, and used it to its benefit also here.

Estimating the annotators behavior. The third experiment studies how *SVGPCR-Mix* exploits the expert labels to learn the annotators behavior. Fig. 7 shows the sensitivity and specificity estimations as the amount of expert labels increases (in the majority of *spammers* scenario). For annotators $1-4$, the estimations keep close to 0.5 (recall from Table 1 that all of them are indeed *spammers*). For annotator 5, whose true sensitivity and specificity values are high, the estimation evolves. In the beginning, *SVGPCR-Mix* cannot distinguish this annotator from the *spammers* due to the identifiability limitation of crowdsourcing methods. However,

[2] Every two expert labels, one is obtained for a sample that also has crowdsourcing labels and the other for a new sample. The relevance of this is analyzed in Fig. 8.

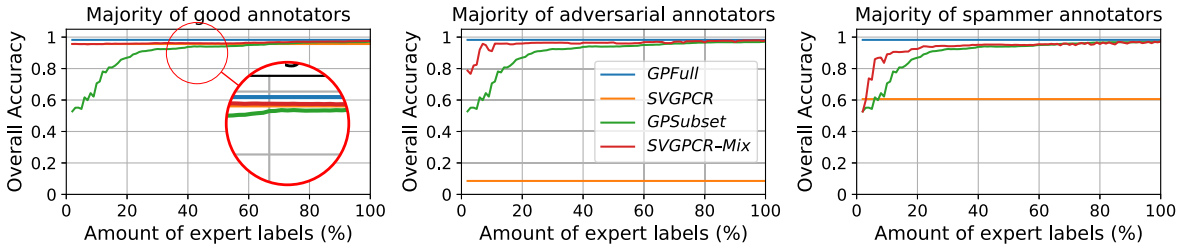

Fig. 5. Performance of *SVGPCR-Mix* and related methods as the amount of expert labels increases in the three different scenarios considered. The proposed fusion of expert labels improves the results, and this is more significant as the annotators are less reliable.

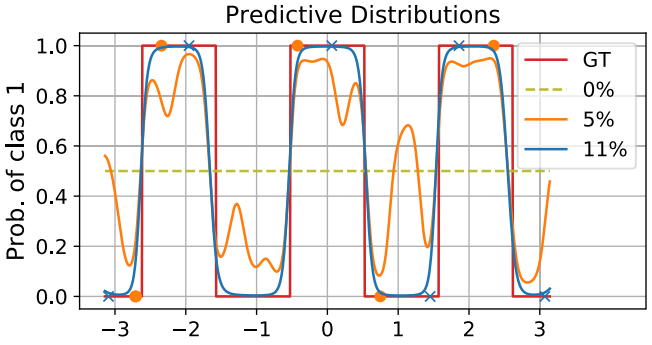

Fig. 6. Predictive distribution of *SVGPCR-Mix* as the amount of expert labels increases. These play the role of anchor points to unravel the ground truth (GT).

as the percentage of expert labels increases, it is able to make a much better estimation.

Analyzing the role of the ELBO's new term. The last experiment studies the influence of the ELBO seventh term (the new one introduced by this model) on the estimation accuracy. As explained in Section 3.2, such term appears if there are samples with both expert and crowdsourcing labels. Therefore, Fig. 8 shows the performance of *SVGPCR-Mix* as expert labels are added following three different schemes: *full* (they are added on samples that also have crowdsourcing labels), *null* (they are added on samples that do not have crowdsourcing labels), and *middle* (every two, one is of the *full*-type and the other of the *null*-type). The results are significantly better for the *full* and *middle* cases, that is, when the seventh term of the ELBO is being actually involved in the calculation, mitigating the identifiability problem.

4.2. Music genre dataset

Motivation and problem formulation. In the previous section, the synthetic dataset allowed for a detailed analysis of *SVGPCR-Mix* behavior. Here we focus on a real-world crowdsourcing problem for which all true labels are available. This allows us to assess the performance of *SVGPCR-Mix* as the amount of expert labels increases in a real-world scenario. Take into account that such analysis will not be possible in the next section (LIGO dataset), where the number of samples with expert labels is only 5% of the number of samples with crowdsourcing labels.

The Music Genre dataset consists of 1000 fragments (30 s length) of songs. The goal is to distinguish between 10 music genres: classical, country, disco, hiphop, jazz, rock, blues, reggae, pop, and metal [39]. For preprocessing and feature extraction, the authors in [40] used the Marsyas music information tool (http://marsyas.info/) to extract 124 features. These features include relevant technical metrics such as means and variances of timbral features, time-domain zero-crossings, spectral centroid, roll-off, flux, and Mel-Frequency Cepstral Coefficients (MFCC).

The dataset contains 100 samples from each genre, which were randomly divided in 70 samples for training and 30 for testing. This results in a total of 700 samples for training and 300 for test (recall that there are ten different genres). Crowdsourcing labels were obtained with Amazon Mechanical Turk, which is one of the most popular crowdsourcing platforms (www.mturk.com). Each annotator listened to a subset of fragments and labeled them as one of the ten genres listed above. A total amount of 2945 labels were provided by 44 different annotators.

***SVGPCR-Mix* outperforms related methods.** Here, *SVGPCR-Mix* is compared to the same methods as in the previous section. The performance in terms of overall accuracy and test likelihood is shown in Fig. 9. Whereas the overall accuracy only considers the predictive mode, the test likelihood also takes into account the quality of the predictive uncertainty. This is an important aspect in practice, where the reliability on the prediction is as important as the prediction itself. The results in Fig. 9 are the mean over ten independent runs.

Interestingly, *SVGPCR-Mix* behaves as theoretically expected. It obtains better results than *SVGPCR* and *GPSubset* in both metrics. Recall that *SVGPCR* and *GPSubset* only leverage one source of information (crowdsourcing labels in the former and expert labels in the latter). Moreover, *SVGPCR-Mix* converges to the golden reference *GPFull* as the amount of expert labels increases. This confirms that, also in a real-world problem, the proposed fusion of crowds and experts provides an empirical benefit.

4.3. Glitch detection in LIGO

In this section, we evaluate the proposed model on the real problem that motivated its development: glitch detection in signals acquired by LIGO.

Problem description. LIGO is a large-scale physics experiment whose goal is to detect gravitational waves (GWs) [41]. GWs are ripples in the fabric of space–time, which are produced by massive astronomical events (such as binary black holes or neutron stars mergers). Although their existence is a theoretical consequence of General Relativity, their first direct observation was made on 2015 by LIGO. The discovery had a tremendous impact in the scientific community, and was awarded the 2017 Nobel in Physics. Specifically, GWs have inaugurated a whole new way to explore the universe, which before could only be perceived through electromagnetic radiation.

To identify GWs, LIGO deploys cutting-edge technology that is sensitive to different sources of noise. This contamination appears as *glitches* in the spectrograms that astrophysicists analyze to search for GWs (recall Fig. 1, which shows examples of two specific types of glitches). The goal of the GravitySpy project[3] is to develop a machine learning system that automatically classifies the different types of glitches. Since LIGO produces a constant stream of data, GravitySpy leverages the Zooniverse platform[4]

[3] https://ciera.northwestern.edu/programs/gravityspy/.

[4] https://www.zooniverse.org/projects/zooniverse/gravity-spy.

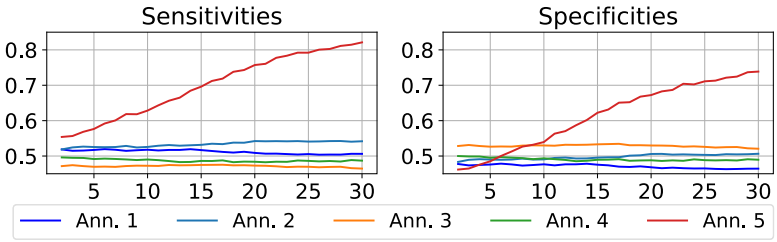

Fig. 7. Sensitivities and specificities estimated by the proposed model in the majority of *spammers* scenario. The *x*-axis shows the number of used expert labels. We observe that *SVGPCR-Mix* leverages expert labels to learn the behavior of annotators.

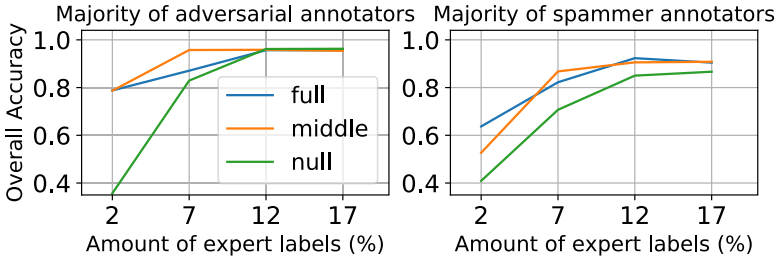

Fig. 8. Performance of *SVGPCR-Mix* as expert labels are added following three different schemes (more details in the text). The best results are obtained when the ELBO seventh term is considered.

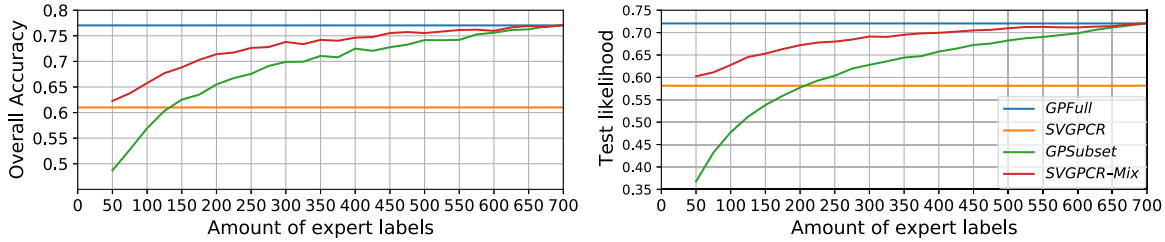

Fig. 9. Performance of *SVGPCR-Mix* and related methods in the real-world Music Genre crowdsourcing dataset as the amount of expert labels used increases. The performance is given in terms of test overall accuracy (left) and test likelihood (right). The proposed *SVGPCR-Mix* obtains better results than *SVGPCR* and *GPSubset*, approaching the golden reference *GPFull* as the amount of expert labels increases.

to obtain crowdsourcing labels. Moreover, to complement these, some expert labels have been provided by astrophysicists.

The GravitySpy dataset. To the best of our knowledge, currently GravitySpy is one of the largest data sets containing crowdsourcing and expert labels. Namely, our training set contains 173,565 samples (glitches) and 1,828,981 crowdsourcing annotations (i.e., a mean value of more than 10 labels per sample), which have been provided by 3443 collaborators through the Zooniverse platform. For each glitch, we use 256 relevant features extracted in [16]. The glitches have been classified into 15 different classes proposed by astrophysicists (they all are shown in [10, Figure 3]). Moreover, there are 7901 samples with expert labels (2593 of them also have crowdsourcing annotations; this ensures that the seventh term of the ELBO is used, recall Fig. 8). GravitySpy test set is made up of 9997 samples.

Baselines. Two methods have addressed this problem so far, and they will be used as baselines for our approach. The first one, which will be referred to as *DL*, uses the expert labels to train a Convolutional Neural Network [16]. The second one is *SVGPCR* [10], which uses a GP-based crowdsourcing model to train with all the crowdsourcing labels. Recall that each one of these methods leverages one type of labels, whereas the proposed *SVGPCR-Mix* is able to train with both. Since *SVGPCR-Mix* is a generalization of both *SVGPCR* and *SVGP* [17] (recall Section 3.2), we also include the later in the comparison for completeness.

***SVGPCR-Mix* achieves state-of-the-art results in the LIGO data.** Table 3 shows the overall accuracy (OA) and test likelihood (TL) for the four compared methods across the different classes. Whereas the former considers just the predictive mode, the latter also takes into account the quality of the uncertainties. *SVGPCR-Mix* consistently obtains the best results in both metrics, which justifies the proposed fusion of expert and crowdsourcing labels. Notice also that the samples with expert labels are only 5% of the samples with crowdsourcing labels. This supports the idea illustrated in the synthetic experiment that just a few expert labels are enough to complement the crowdsourcing ones.

Let us analyze several aspects of the performance more in-depth. The good performance of *SVGP* and *DL* (which only use 7901 samples) is due to (1) the quality of the expert labels and (2) (for *DL*) the representation power of Convolutional Neural Nets of spectrograms (images). Notice also that the results of *SVGPCR* are not far from those of *SVGPCR-Mix*. This implies that most annotators are reliable (otherwise, the identifiability issues would severely harm the performance of *SVGPCR*, recall Fig. 5). This is a valuable piece of information for astrophysicists, since it validates the training system designed for the volunteers. We further verify it by empirically estimating the overall accuracy of annotators. We do it based on the 2593 samples that have both expert and crowdsourcing labels. Indeed, results in Fig. 10 show an estimated OA greater than 0.9 for almost all the annotators. Finally, we also stress the scalability of *SVGPCR-Mix*, which is able to cope with 173 565 training samples and 1 828 981 crowdsourcing labels (far beyond the standard GPs limit).

Confusion matrix estimation. Next, let us illustrate the ability of *SVGPCR-Mix* to estimate the annotators confusion matrices. We consider annotator #80, which has annotated many samples that

Table 3
Performance (accuracy and test likelihood) for the four compared methods in the LIGO problem. *DL* is the Convolutional Neural Network introduced in [16], *SVGP* refers to the Scalable Variational GP introduced in [17], and *SVGPCR* denotes the crowdsourcing method in [10]. The proposed *SVGPCR-Mix* obtains the best global results in terms of both OA and TL.

Classes	Overall accuracies				Test likelihood			
	DL	*SVGP*	*SVGPCR*	*SVGPCR-Mix*	*DL*	*SVGP*	*SVGPCR*	*SVGPCR-Mix*
1080LINE	.9759 (.0025)	.9697 (.0064)	.9720 (.0069)	**.9883 (.0023)**	.9727 (.0023)	.9209 (.0088)	.9688 (.0075)	**.9853 (.0021)**
1400RIPPLE	.7569 (.0106)	.6642 (.0416)	**.8577 (.0171)**	.7967 (.0267)	.7541 (.0064)	.5884 (.0363)	**.8509 (.0156)**	.7975 (.0275)
BLIP	.9603 (.0018)	.9592 (.0032)	.9622 (.0052)	**.9715 (.0028)**	.9587 (.0012)	.9481 (.0057)	.9587 (.0055)	**.9685 (.0024)**
EXTR.LOUD	.8136 (.0185)	**.8784 (.0283)**	.7295 (.0427)	.8273 (.0422)	.7993 (.0156)	.7835 (.0225)	.7242 (.0408)	**.8146 (.0435)**
KOIFISH	.7797 (.0132)	.7992 (.0125)	**.8828 (.0115)**	.8522 (.0127)	.7711 (.0112)	.7788 (.0130)	**.8784 (.0117)**	.8484 (.0138)
L.F.BURST	**.8996 (.0052)**	.8904 (.0115)	.8861 (.0105)	.8983 (.0057)	**.8988 (.0056)**	.8787 (.0128)	.8838 (.0098)	.8959 (.0053)
L.F.LINE	.8490 (.0152)	.8693 (.0304)	**.9156 (.0111)**	.8785 (.0132)	.8403 (.0144)	.8304 (.0326)	**.9118 (.0103)**	.8752 (.0135)
NOGLITCH	.9290 (.0025)	.9400 (.0068)	.7951 (.0162)	**.9506 (.0071)**	.9272 (.0019)	.8791 (.0139)	.7932 (.0146)	**.9461 (.0055)**
OTHER	.4859 (.0141)	**.4954 (.0212)**	.4011 (.0091)	.3870 (.0132)	**.4800 (.0119)**	.4571 (.0137)	.3999 (.0091)	.3854 (.0155)
P.L.60HZ	.7983 (.0165)	.9264 (.0076)	.8425 (.0127)	**.9396 (.0037)**	.7937 (.0181)	.8720 (.0135)	.8380 (.0107)	**.9374 (.0055)**
REP.BLIPS	.5197 (.0137)	.5581 (.0509)	**.6700 (.0210)**	.6641 (.0289)	.5227 (.0087)	.5094 (.0432)	**.6651 (.0198)**	.6493 (.0223)
SCATT.LIGHT	.9585 (.0016)	.9580 (.0071)	.9562 (.0056)	**.9667 (.0024)**	.9581 (.0011)	.9302 (.0107)	.9520 (.0057)	**.9640 (.0024)**
SCRATCHY	**.9220 (.0060)**	.9013 (.0148)	.9000 (.0165)	.8847 (.0166)	**.9194 (.0055)**	.8419 (.0107)	.8953 (.0176)	.8819 (.0204)
VIOLIN	.9769 (.0013)	.9700 (.0032)	**.9914 (.0017)**	.9758 (.0016)	.9764 (.0006)	.9574 (.0045)	**.9886 (.0014)**	.9738 (.0011)
WHISTLE	.9535 (.0069)	**.9649 (.0000)**	.9201 (.0047)	.9535 (.0111)	**.9528 (.0019)**	.9370 (.0037)	.9179 (.0046)	.9483 (.0060)
GLOBAL	.9113 (.0020)	.9145 (.0043)	.9183 (.0027)	**.9258 (.0019)**	.9081 (.0018)	.8813 (.0045)	.9149 (.0027)	**.9227 (.0073)**

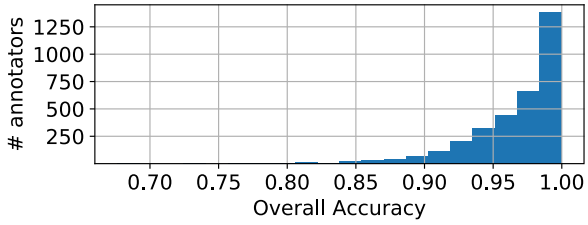

Fig. 10. Histogram of the annotators according to their Overall Accuracy evaluated on $\mathbf{Z}_{\mathcal{O}}$. In this problem there exists a majority of good annotators.

also have expert labels (namely, 927 of them). This allows us to empirically calculate its confusion matrix through a frequentist analysis of its annotations. Additionally, we only consider the classes for which the selected annotator provided more than 100 annotations. The confusion matrices estimated by *SVGPCR-Mix* (and also by *SVGPCR*) are shown at the top of Fig. 11. Both matrices are very similar, and have values close to the empirical estimation. Notice also that the annotator is a reliable one (matrices do not look diagonal because only classes with at least 100 annotations are shown). Recall from Section 3.2 that annotator confusion matrices are estimated through a posterior Dirichlet distribution q(**R**). The value reported here is the expectation of this distribution.

In the second row of Fig. 11, we compare the confusion vectors for classes 3 (BLIP) and 12 (SCATTERED LIGHT). By confusion vector we refer to a column of the confusion matrix, i.e., the probabilities assigned by the annotator for a certain class. Here we chose these two classes for being those where *SVGPCR-Mix* and *Empirical* confusion vectors are most similar and different (in the squared error sense), respectively. However, in both cases we observe that *SVGPCR* and *SVGPCR-Mix* almost match the empirical value, which confirms the accuracy of their estimations. In addition to the discussed case of annotator #80, the global examination of *SVGPCR* and *SVGPCR-Mix* confusion matrices yields very similar results. This makes us conclude that the improvement in performance is due to a better underlying classifier, which benefits from the proposed fusion of expert and crowdsourcing labels.

Limitations of the method. Finally, to provide a deeper analysis of the proposed method, it is worth discussing potential limitations (as well as possible solutions). For instance, notice that *SVGPCR-Mix* does not perform feature extraction on its own, as it is fed with raw data or previously extracted features (as in the case of GravitySpy). This limits its direct application on highly structured data such as images or audio. An interesting line of future research is to leverage deep kernel learning (DKL) techniques so that the kernel used in *SVGPCR-Mix* allows for feature extraction. Another limitation is that the interpretability of the estimated inducing point locations is low, due to their high dimension. This limitation is inherited from the sparse GP theory, and could be addressed with techniques from that field, see e.g. [42]. Also, dimensionality reduction methods can help to make inducing point locations more interpretable.

5. Conclusions

In this work we have proposed a new probabilistic model for detecting glitches in signals acquired by LIGO. The dataset collected by the GravitySpy project motivated the development of an algorithm combining the quality of labels provided by experts with the ability of the crowds to label huge data sets. The proposed method is a natural generalization of *SVGPCR* and *SVGP*. We have studied the identifiability issues of standard crowdsourcing methods. We have demonstrated that the use of true labels makes our method robust in scenarios where the majority of annotators are not reliable, whereas previous crowdsourcing methods in the literature catastrophically fail in this case. Furthermore, we have seen that only a small percentage of samples with true labels suffices for *SVGPCR-Mix* to recognize the behavior of annotators and extract all the useful knowledge from the crowdsourcing data. We have subsequently applied *SVGPCR-Mix* to the GravitySpy data, establishing a new state-of-the-art approach for this problem. Finally, we have illustrated the differences between *SVGPCR* and *SVGPCR-Mix* when estimating the confusion matrices of GravitySpy annotators.

This work is a relevant contribution in the growing field of citizen science, as it allows for a smarter collaboration between its two main actors: crowds and experts. More importantly, our results suggest that the participation in citizen science projects could be extended to wider pools when using *SVGPCR-Mix*, since expert labels have proven especially useful in scenarios with plenty of noise (i.e. those with a majority of spammer and adversarial annotators). We hope that the proposed method fosters new research in this direction, making a real impact on other ambitious crowdsourcing projects in addition to gravitational waves search.

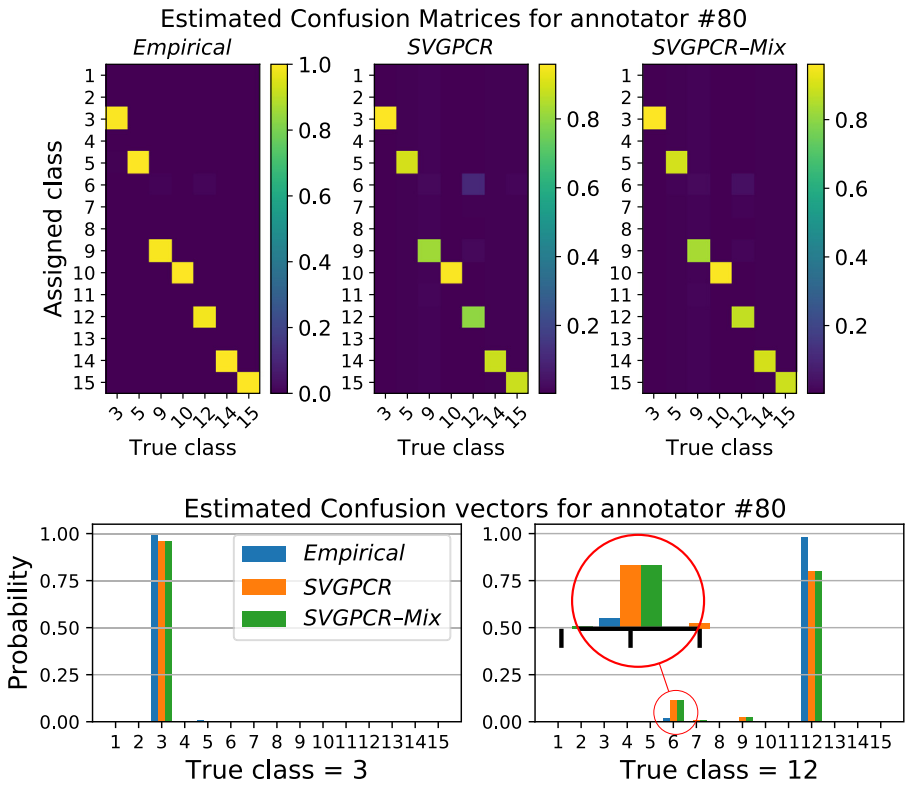

Fig. 11. First row, from left to right: for certain annotator, *Empirical* confusion matrix and those estimated by *SVGPCR* and *SVGPCR-Mix*, respectively. Second row: for the same annotator, detail of the assigned classes for two different true classes (BLIP and SCATTERED LIGHT). These are commonly referred to as confusion vectors. The estimations by both *SVGPCR* and *SVGPCR-Mix* are very similar to the empirical values.

CRediT authorship contribution statement

Pablo Ruiz: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Visualization. **Pablo Morales-Álvarez:** Conceptualization, Methodology, Data curation, Formal analysis, Writing – original draft, Visualization. **Scott Coughlin:** Resources, Data curation, Writing – review & editing. **Rafael Molina:** Conceptualization, Methodology, Formal analysis, Resources, Writing – review & editing, Supervision, Funding acquisition. **Aggelos K. Katsaggelos:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Halevy, P. Norvig, F. Pereira, The unreasonable effectiveness of data, IEEE Intell. Syst. 24 (2) (2009) 8–12.
[2] D. Brickley, M. Burgess, N. Noy, Google Dataset Search: Building a search engine for datasets in an open Web ecosystem, in: The World Wide Web Conference, 2019, pp. 1365–1375.
[3] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, J.R. McClean, Power of data in quantum machine learning, Nature Commun. 12 (1) (2021) 1–9.
[4] H. Ibrahim, X. Liu, N. Zariffa, A.D. Morris, A.K. Denniston, Health data poverty: an assailable barrier to equitable digital health care, in: The Lancet Digital Health 2021, Elsevier, 2021.
[5] E. Salarioglu, O. Gungor, Crowdsourcing in remote sensing: A review of applications and future directions, IEEE Geosci. Remote Sens. Mag. 8 (4) (2020) 89–110.
[6] Y. Wu, A. Schmidt, E. Hernández-Sánchez, R. Molina, A.K. Katsaggelos, Combining attention-based multiple instance learning and Gaussian processes for CT hemorrhage detection, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2021, pp. 582–591.
[7] A. Irwin, No PhDs needed: how citizen science is transforming research, Nature 562 (2018) 480–482.
[8] M. López-Pérez, M. Amgad, P. Morales-Álvarez, P. Ruiz, L.A. Cooper, R. Molina, A.K. Katsaggelos, Learning from crowds in digital pathology using scalable variational Gaussian processes, Sci. Rep. 11 (1) (2021) 1–9.
[9] A.N. Uma, T. Fornaciari, D. Hovy, S. Paun, B. Plank, M. Poesio, Learning from disagreement: A survey, J. Artificial Intelligence Res. 72 (2021) 1385–1470.
[10] P. Morales-Álvarez, P. Ruiz, S. Coughlin, R. Molina, A.K. Katsaggelos, Scalable variational Gaussian processes for crowdsourcing: Glitch detection in LIGO, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2022) 1534–1551, http://dx.doi.org/10.1109/TPAMI.2020.3025390.
[11] L. Zhang, R. Tanno, M.-C. Xu, C. Jin, J. Jacob, O. Cicarrelli, F. Barkhof, D. Alexander, Disentangling human error from ground truth in segmentation of medical images, Adv. Neural Inf. Process. Syst. 33 (2020) 15750–15762.
[12] F. Tao, L. Jiang, C. Li, Differential evolution-based weighted soft majority voting for crowdsourcing, Eng. Appl. Artif. Intell. 106 (2021) 104474.
[13] A. Dawid, A. Skene, Maximum likelihood estimation of observer error-rates using the EM algorithm, J. Real Stat. Soc. Ser. C (Appl. Stat.) 28 (1) (1979) 20–28.
[14] M. Zevin, S. Coughlin, S. Bahaadini, N. Besler, S. Allen, et al., Gravity Spy: integrating advanced LIGO detector characterization, Machine Learning, and citizen science, Classical Quantum Gravity 34 (6) (2017) 064003.
[15] Z. Ramezani, A. Pourdarvish, Transfer learning using tsallis entropy: An application to gravity spy, Phys. A 561 (2021) 125273.
[16] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J. Smith, V. Kalogera, A. Katsaggelos, Machine learning for gravity spy: Glitch classification and dataset, Inform. Sci. 444 (2018) 172–186.
[17] J. Hensman, A. Matthews, Z. Ghahramani, Scalable variational Gaussian process classification, in: International Conference on Artificial Intelligence and Statistics (AISTATS), Vol. 38, 2015, pp. 351–360.
[18] D. Blei, A. Kucukelbir, J. McAuliffe, Variational inference: A review for statisticians, J. Amer. Statist. Assoc. 112 (518) (2017) 859–877.
[19] P. Ruiz, P. Morales-Álvarez, R. Molina, A. Katsaggelos, Learning from crowds with variational Gaussian processes, Pattern Recognit. 88 (2019) 298–311, http://dx.doi.org/10.1016/j.patcog.2018.11.021.
[20] C. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[21] C. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT, 2006.
[22] E. Snelson, Z. Ghahramani, Sparse Gaussian Processes using pseudo-inputs, in: Advances in Neural Information Processing Systems (NIPS), 2006, pp. 1257–1264.
[23] J. Hensman, N. Fusi, N.D. Lawrence, Gaussian processes for big data, in: Conference on Uncertainty in Artificial Intelligence (UAI), 2013, pp. 282–290.
[24] J. Whitehill, T.-F. Wu, J. Bergsma, J.R. Movellan, P.L. Ruvolo, Whose vote should count more: Optimal integration of labels from labelers of unknown expertise, in: Advances in Neural Information Processing Systems (NIPS), 2009, pp. 2035–2043.
[25] P.G. Ipeirotis, F. Provost, J. Wang, Quality management on amazon mechanical turk, in: ACM SIGKDD Workshop on Human Computation (HCOMP'10), 2010, pp. 64–67.
[26] V. Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Hermosillo Valadez, L. Bogoni, 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, ACM, 2009, pp. 889–896.
[27] V. Raykar, S. Yu, L. Zhao, G. Hermosillo Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, J. Mach. Learn. Res. 11 (Apr) (2010) 1297–1322.
[28] F. Rodrigues, F. Pereira, B. Ribeiro, Gaussian Process classification and active learning with multiple annotators, in: International Conference on Machine Learning (ICML), 2014, pp. 433–441.
[29] T.P. Minka, A Family of Algorithms for Approximate Bayesian Inference (Ph.D. thesis), University of Cambridge, 2001.
[30] P. Morales-Álvarez, P. Ruiz, R. Santos-Rodríguez, R. Molina, A. Katsaggelos, Scalable and efficient learning from crowds with Gaussian processes, Inf. Fusion 52 (2019) 110–127, http://dx.doi.org/10.1016/j.inffus.2018.12.008.
[31] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, N. Navab, AggNet: Deep learning from crowds for mitosis detection in breast cancer histology images, IEEE Trans. Med. Imaging 35 (5) (2016) 1313–1321.
[32] F. Rodrigues, F. Pereira, Deep learning from crowds, in: Conference on Artificial Intelligence (AAAI), 2018, pp. 1611–1618.
[33] K. Murphy, Machine Learning: A Probabilistic Perspective, MIT, 2012.
[34] D. Hernández-Lobato, J. Hernández-Lobato, P. Dupont, Robust multi-class Gaussian process classification, in: Advances in Neural Information Processing Systems (NIPS), 2011, pp. 280–288.
[35] P. Morales-Alvarez, A. Pérez-Suay, R. Molina, G. Camps-Valls, Remote sensing image classification with large-scale Gaussian processes, IEEE Trans. Geosci. Remote Sens. 56 (2) (2017) 1103–1114.
[36] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference for Learning Representations (ICLR), 2015.
[37] A.G.d.G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, J. Hensman, GPflow: A Gaussian process library using TensorFlow, J. Mach. Learn. Res. 18 (40) (2017) 1–6, URL http://jmlr.org/papers/v18/16-537.html.
[38] M. Titsias, Variational learning of inducing variables in sparse Gaussian Processes, in: International Conference on Artificial Intelligence and Statistics (AISTATS), Vol. 5, 2009, pp. 567–574.
[39] G. Tzanetakis, P. Cook, Musical genre classification of audio signals, IEEE Trans. Speech Audio Process. 10 (5) (2002) 293–302.
[40] F. Rodrigues, F. Pereira, B. Ribeiro, Learning from multiple annotators: Distinguishing good from random labelers, Pattern Recognit. Lett. 34 (12) (2013) 1428–1436.
[41] A. Abramovici, W. Althouse, R.P. Drever, Y. Gürsel, S. Kawamura, F. Raab, D. Shoemaker, L. Sievers, R. Spero, K. Thorne, R. Vogt, R. Weiss, S. Whitcomb, M. Zucker, LIGO: The laser interferometer gravitational-wave observatory, Science 256 (5055) (1992) 325–333, http://dx.doi.org/10.1126/science.256.5055.325, arXiv:http://science.sciencemag.org/content/256/5055/325.full.pdf.
[42] H. Liu, Y.-S. Ong, X. Shen, J. Cai, When Gaussian process meets big data: A review of scalable GPs, IEEE Trans. Neural Netw. Learn. Syst. 31 (11) (2020) 4405–4423.