Non-parametric Bayesian annotator combination

Maximilien Servajean, Romain Chailan, Alexis Joly

Non-parametric Bayesian Annotator Combination

M. Servajean[a,b,c,*], R. Chailan[d], A. Joly[b]

[a] *Université Paul Valéry Montpellier, Route de Mende 34199 Montpellier, France.*
[b] *Zenith Team from INRIA at LIRMM, 860 rue de St Priest, 34095 Montpellier, France.*
[c] *ADVANSE Team at LIRMM.*
[d] *Twin Solutions, 11, rue Dulong, 75017 Paris, France.*

Abstract

Relying on a single imperfect human annotator is not recommended in real crowdsourced classification problems. In practice, several annotators' propositions are generally aggregated to obtain a better classification accuracy.

Bayesian approaches, by modelling the relationship between each annotator's output and the possible true labels (classes), have been shown to outperform other simpler models.

Unfortunately, they assume that the total number of true labels is known. This is not the case in lots of realistic scenarios such as open-world classification where the number of possible labels is undetermined and may change over time.

In this paper, we show how to set a non-parametric prior over the possible label set using the Dirichlet process in order to overcome this limitation. We illustrate this prior over the Bayesian annotator combination (BAC) model from the state of the art, resulting in the so-called non-parametric BAC (NPBAC).

*Corresponding author
Email address: servajean@lirmm.fr (M. Servajean)

We show how to derive its variational equations to evaluate the model and how to assess it when the Dirichlet process has a prior using the Laplace method.

We apply the model to several scenarios related to closed-world classification, open-world classification and novelty detection on a dataset previously published and on two datasets related to plant classification. Our experiments show that NPBAC is able to determine the true number of labels, but also and surprisingly, it largely outperforms the parametric annotator combination by modelling more complex confusions, in particular when few or no training data are available.
Keywords: Bayesian, variational, Laplace, classification, combination, Dirichlet process, crowdsourcing

---

1. Introduction

The huge potential of leveraging human power has been noted in recent years, especially when typical machine learning techniques such as deep learning [23] fails. This is particularly needed for instance when constructing the large training sets needed by these techniques [9]. In this paper, we focus on the particular case of data classifications, *i.e.* given a set of data items such as images, sounds or any other documents, we would like to associate a label to each of them. Unfortunately, a perfect annotator that would obtain 100% accuracy does not exist in most realistic scenarios. In practice, we often aggregate information from several annotators, hoping that their abilities are complementary and that the resulting aggregation has better accuracy than a single annotator [23]. Crowdsourcing platforms, such as Amazon Mechanical

Turk or Zooniverse, offer an efficient way to involve lots of annotators and collect their classification propositions [6, 24]. Similarly, in a previous work, we presented The Plant Game, a gamified approach to crowdsourcing where each classification proposition given by an annotator and validated by the crowdsourced consensus increases the annotator's ranking [21]. In all of these platforms, annotators are asked to propose a label without the knowledge of the propositions put forward by other annotators.

A common problem is therefore to merge classification propositions. A simple approach involves counting the number of times each label (class) has been proposed which is called majority voting. More sophisticated methods can be devised such as weighed majority voting [16], where the weight of each annotator depends on its overall classification accuracy. In these approaches, the label of each item stems from the classification propositions. Recent studies have focused on the Bayesian combination of so-called imperfect annotators' propositions [15, 17, 23]. These models rely on the idea of *confusions* which consists of modelling the output probability of each annotator given all possible true labels – two annotators can have two totally different outputs, such as humans speaking different languages. Therefore they outperform the majority voting approaches. As an example, let us consider a scenario based on ImageNet [9]. The ImageNet challenge consists of finding the true class of a set of images over a large variety of values, such as cars, flowers, etc. An annotator's confusion could indicate that he/she is not capable of disambiguating different types of flowers.

Unfortunately, and contrary to simple majority voting, Bayesian approaches do assume that the number of true labels is known when computing

all confusion matrices. This assumption is strong and can be unrealistic in some scenarios. In open-world classification problems [2], determining the set of possible true labels is impossible and can even change over time. In biodiversity surveillance on a crowdsourcing platform, the annotators have to identify species of plants based on their images and they are particularly interested in detecting new species. This would not be possible with a fixed predetermined number of true labels.

In this paper, we propose a non-parametric Bayesian combination model to solve the problem of combining annotators' propositions when the label set is not initially known. In addition, we will show that a non-parametric model enables us to take more complex confusions for each annotator into account.

Related studies focused on Bayesian non-parametric models often rely on a distribution called the Dirichlet process [4, 10]. The basic intuition behind such distributions is that the number of possible labels is theoretically infinite while several data items (*e.g.* images, sounds) can have the same label with a positive probability. More formally, the Dirichlet process has infinite dimensions while almost surely staying discrete[1]. The granularity of each resulting class of the non-parametric model depends on the concentration parameter of the Dirichlet process.

However, even though fixing the concentration parameter is less problematic than having to fix the number of possible labels, we also study the model when the concentration parameter itself follows a prior distribution. Thus,

[1] "almost surely" refers to the fact that some outcomes, while being theoretically possible, have a zero probability.

the model should converge to the "best" granularity based on the observed data and our prior knowledge.

In order to infer the posterior probabilities, we derive all variational equations required by the model. Variational inference [4, 23] is known to approximate the joint probability very efficiently while sampling based methods are known to be much slower [3, 23]. Unfortunately, setting a prior over the concentration parameter makes its variational equation intractable. To solve this issue, we show that the Laplace development of the concentration parameter variational equation approximates it by a Gaussian distribution.

In summary, this paper introduces the following original contributions:

- We propose a non-parametric Bayesian annotator combination model to solve the problem of learning the model when the labels set is not known as well as the problem of modelling complex confusions. We also discuss its relationship with the classical parametric model (described in [15, 23]).

- We develop variational equations of the non-parametric model in order to efficiently estimate its joint probability, even in high dimensions.

- We show how the Dirichlet process parameter itself can be described with a distribution and how to compute its variational equation using the Laplace method.

- We present an extensive application analysis of previous contributions in the experiments section and show that NPBAC can correctly estimate the number of classes and even outperforms the state of the

art Bayesian combination approach that we build upon as well as the simpler majority voting approach.

The rest of this paper is structured as follows. The related work is introduced in Section 2. Section 3 describes the classical parametric Bayesian annotator combination model. In Section 4, we show how to transform the parametric model with the Dirichlet process to make it non-parametric. The variational equations of the non-parametric model are explained in Section 5. In Section 6, we show how to add a prior distribution over the concentration parameter and how to estimate its posterior distribution with the Laplace development of its variational equation. In Section 7 we report and discuss the results of our experiments.

2. Related Work

Whereas crowdsourcing is a relatively new domain [6, 11, 14], contributions related to human classifiers (*i.e.* annotators) combinations or error-rate evaluations go back as far as the 1970*s*. Dawid and Skene [8], in particular, focused on estimating the error-rate of several expert annotators from a noisy ground truth. This paper underlies several recent works that we will present in this section.

In [25], Tulyakov *et al.* propose an overview of several classifier combination methods: from the simplest, such as majority voting or Borda – the rank of a result depends on its rank proposed by all classifiers – to more complex methods such as those using the Dempster-Shafer theory of evidence.

Recently, Bayesian models have been shown to outperform other combination methods [15, 20, 23]. In [20], Raykar *et al.* show how to train a

machine learning classifier using data features and classification votes from human annotators. Their approach is only partially Bayesian as the model is simply used as a "point estimate". Moreover, the main goal was not to perform classification but to learn a machine learning classifier on more realistic labels: each item in the training dataset is not associated with a single class but rather to a probability distribution over all classes. The goal is to determine these probability distributions and to train a classifier on it.

Kim and Ghahramani [15] propose a complete Bayesian model to aggregate the classification votes of multiple imperfect human or machine classifiers. They develop two main approaches, one taking the dependency among classifiers into account and one ignoring it. They show that both models actually achieve very similar results. In addition, they use Gibbs sampling to evaluate the model.

In [26], Venanzi *et al.* propose to exploit correlations among annotators to increase the classification quality. Each user is thus part of a community whose members likely have the same confusion matrix. The main advantage of this approach is to estimate the classification capabilities of the annotators more precisely even when very few classification examples are available.

Simpson *et al.* [23] propose a Bayesian approach similar to that introduced by Kim and Ghahramani. However, they use variational Bayes to estimate the joint probability. Moreover, they model evolving skills using the dynamic classifier combination notion. Their variational model underlies our inference procedure.

Welinder and Perona [28] also provide a Bayesian approach for image classification. In addition to combining multiple annotators, they focus on

task assignment using the inferred abilities of each one of them.

Bragg *et al.* [5] propose a method where the confidence in an annotator's ability is computed based on his/her number of correct classifications (from a ground truth). In our study, we use a Bayesian network to infer annotators' abilities.

Moreno *et al.* [17] propose a non-parametric approach for aggregating votes. The authors focus on crowdsourcing and propose to cluster similar users when few votes are available. The cluster set has a non-parametric prior and their number grows with the data. An annotator's confusion matrix has a prior based on the cluster he/she belongs to. The non-parametric aspect of their model is therefore in the number of clusters and they show that parametric and non-parametric clustering obtain similar results. Note however that their model is parametric in the number of labels which still need to be set *a priori*.

However, contrary to the presented model, all of these studies assume that the label set is either known or can easily be estimated from the observed data (because there is a bijection between the annotator/classifier output and the true label set). In this paper, we study the NPBAC model to solve this problem. We illustrate how to make the number of labels non-parametric and we apply our contribution to the approaches presented in [15] and [23] but the same is applicable to other Bayesian models.

3. Bayesian Annotator Combination (Baseline)

In this section we discuss the particular parametric case of the Bayesian classifier/annotator combination (BAC) approach used in [23] and presented

in Figure 1. A typical common assumption is to consider all annotators as independent. Kim and Ghahramani [15] have shown that although it is possible to model the dependency among annotators, this does not actually enhance the model accuracy but only its computational needs.

In the Bayesian annotator combination approach, we are given a set of N items where the i^{th} item has the unknown true label t_i. It is assumed that the possible labels are known and indexed from 1 to J, so that $t_i \in \{1, ..., J\}$. Then t_i is considered to be generated by a multinomial distribution of parameters $\boldsymbol{\kappa}$: $p(t_i = j|\boldsymbol{\kappa}) = \kappa_j$. We are then given K annotators which produce finite outputs indexed by $1, ..., L$, where L is the total number of possible outputs. In particular, $c_i^{(k)}$ refers to the output of annotator k for the data item indexed by i. Let us assume that $c_i^{(k)}$ follows a multinomial distribution of parameters $\boldsymbol{\pi}_{\boldsymbol{t_i}}^{(\boldsymbol{k})}$: $p(c_i^{(k)} = l|t_i = j, \boldsymbol{\pi}_{\boldsymbol{j}}^{(\boldsymbol{k})}) = \pi_{jl}^{(k)}$. The literature generally refers to $\boldsymbol{\pi}^{(\boldsymbol{k})}$ as the confusion matrix of annotator k; basically, given a true label j, $\boldsymbol{\pi}_{\boldsymbol{j}}^{(\boldsymbol{k})}$ refers to the confusion of annotator k for this specific label. Both $\boldsymbol{\pi}_{\boldsymbol{j}}^{(\boldsymbol{k})}$ and $\boldsymbol{\kappa}$ are considered to be drawn from Dirichlet distributions of parameters $\boldsymbol{\alpha}_{\boldsymbol{0,j}}^{(\boldsymbol{k})}$ and $\boldsymbol{\nu_0}$, respectively. Note that $\boldsymbol{\alpha}$ and $\boldsymbol{\nu}$ are indexed by 0 acknowledging they are our prior belief. Inferring the model means that we will update our belief over the variables by taking into account the observable variables, that is, the annotator outputs.

As explained in Simpson *et al.* [23], the joint probability for this model is expressed as

$$p(\boldsymbol{\kappa}, \boldsymbol{\Pi}, \boldsymbol{t}, \boldsymbol{c}|\boldsymbol{A_0}, \boldsymbol{\nu}) = (\prod_{i=1}^{N} \kappa_{t_i} \prod_{k=1}^{K} \pi_{t_i, c_i^{(k)}}^{(k)}) p(\boldsymbol{\kappa}|\boldsymbol{\nu_0}) p(\boldsymbol{\Pi}|\boldsymbol{A_0}), \tag{1}$$

where $\boldsymbol{\Pi}$ refers to all confusion matrices $\boldsymbol{\pi}^{(\boldsymbol{k})}$ and $\boldsymbol{A_0}$ refers to all parameters

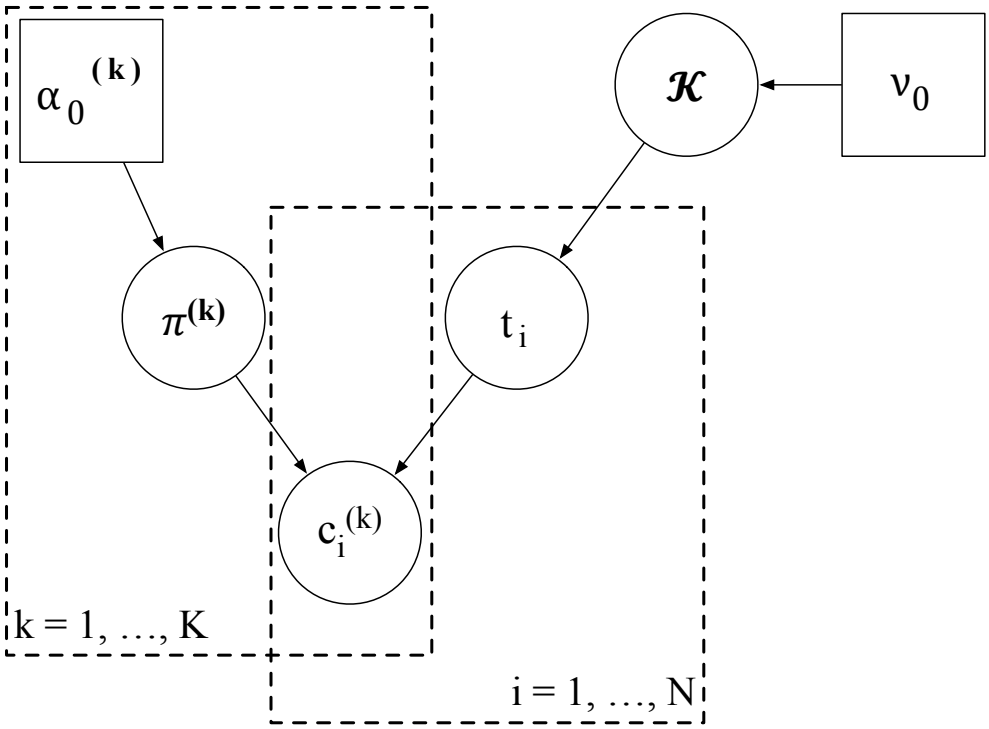

Figure 1: Parametric Bayesian annotator combination.

$\boldsymbol{\alpha}_{\mathbf{0}}^{(k)}$ for each annotator k. It is assumed in this model that the number of different true labels J is known in advance (e.g. matrices $\pi^{(k)}$ are of fixed size $J \times L$). This assumption does not necessarily exist in real life. In the sequel we investigate a non-parametric approach to avoid having to set J. This approach relies on the Dirichlet process [10].

4. Dirichlet Process

In subsection 4.1, we first describe how the Dirichlet process stems from the definition of the parametric BAC model introduced in the previous section. Then, in subsection 4.2, we present the so-called "stick-breaking" rep-

resentation of the Dirichlet process and how it is integrated in our model.

4.1. From the Dirichlet Distribution to the Dirichlet Process

The intuition of the Dirichlet process introduced by Ferguson [10] stems from a multinomial distribution and its conjugate prior, *i.e.* the Dirichlet distribution. Recall that for any data item indexed by i, variable t_i denotes its true label and is drawn from a multinomial distribution of parameter $\boldsymbol{\kappa}$. Thus,

$$p(t_1, ..., t_N|\boldsymbol{\kappa}) = \prod_{j=1}^{J} \kappa_j^{n_j},$$

where n_j refers to the number of data items with true label j. In addition, $\boldsymbol{\kappa}$ is drawn from a Dirichlet distribution of parameter $\boldsymbol{\nu}$. Let us suppose that our prior belief for $\boldsymbol{\kappa}$ is uninformative, leading to

$$\nu_1 = ... = \nu_J = \frac{\beta}{J},$$

where β/J can be seen as a pseudo-count for each class j. We obtain the following probability distribution:

$$\boldsymbol{\kappa}|\boldsymbol{\nu} \sim Dir(\frac{\beta}{J}, ..., \frac{\beta}{J}) = \frac{\Gamma(\beta)}{\Gamma(\beta/J)^J} \prod_{j=1}^{J} \kappa_j^{\beta/J-1}.$$

Let us integrate out $\boldsymbol{\kappa}$ (express $t_1, ..., t_N$ with respect to β/J)[2]:

[2]Thanks to the following trick which consists of integrating a Dirichlet distribution over all possible values:

$$\frac{\Gamma(\sum_{j=1}^{J} \beta_j)}{\prod_{j=1}^{J} \Gamma(\beta_j)} \int \prod_{j=1}^{J} \kappa_j^{\beta_j - 1} d\boldsymbol{\kappa} = 1$$

$$p(t_1, ..., t_N|\beta) = \int p(t_1, ..., t_N|\boldsymbol{\kappa}) p(\boldsymbol{\kappa}|\beta) d\boldsymbol{\kappa}$$

$$= \frac{\Gamma(\beta)}{\Gamma(\beta/J)^J} \int \prod_{j=1}^{J} \kappa_j^{\beta/J-1+n_j} d\boldsymbol{\kappa}$$

$$= \frac{\Gamma(\beta)}{\Gamma(\beta+N)} \prod_{j=1}^{J} \frac{\Gamma(\beta/J+n_j)}{\Gamma(\beta/J)}.$$

The probability of the true label of the i^{th} item conditioned over the other items can thus be expressed as follows:

$$p(t_i = j|\bar{t_i}, \beta) = \frac{n_{\bar{t_i},j} + \beta/J}{N-1+\beta},$$

where $\bar{t_i}$ denotes $\{t_m : m \in 1, ..., N, m \neq i\}$ and $n_{\bar{t_i},j}$ is the number of items that have true label j without counting t_i. In other words , the i^{th} item is more likely to belong to a class where there already are a lot of data items. Now, taking the limit of J to infinity yields the Dirichlet process:

$$p(t_i = j|\bar{t_i}, \beta) = \begin{cases} \frac{n_{\bar{t_i},j}}{N-1+\beta}, & j \in \{1, ..., R\} \text{ if } t_i \text{ takes an existing value,} \\ \frac{\beta}{N-1+\beta}, & \text{if } t_i \text{ takes a new value } (i.e., n_{\bar{t_i},j} = 0). \end{cases}$$

Here R is the number of classes j with $n_{\bar{t_i},j} > 0$. Note that $\beta/(N-1+\beta) = 1 - \sum_{j=1}^{R} n_{\bar{t_i},j}/(N-1+\beta)$. That is, the true label of the i^{th} item can either take a new value (*i.e.* a new true label) with a probability proportional to β, or an existing value with the probability of each existing true label proportional to the number of items already associated to it. Note that the

---

$$\frac{\prod_{j=1}^{J} \Gamma(\beta_j)}{\Gamma(\sum_{j=1}^{J} \beta_j)} = \int \prod_{j=1}^{J} \kappa_j^{\beta_j - 1} d\boldsymbol{\kappa}.$$

distribution remains discrete even though the number of possible classes is infinite. Note also that the order in which the items are considered does not affect the resulting joint distribution probability. Parameter β is known as the scaling parameter (*i.e.* concentration parameter). A bigger value will force the final classification to have smaller granularity, while a smaller value will force the opposite.

4.2. Stick-Breaking Representation

Another way of describing the Dirichlet process is through a stick-breaking representation, as introduced by Sethuraman [22]. First, let us consider that all possible true labels are ordered by their index j. Then consider an infinite collection of random variables $\boldsymbol{\nu}$, where each ν_j represents the probability that an item will be associated with the j^{th} true label given that it is not associated with the $j-1$ first labels. Each ν_j is assumed to be generated by a beta distribution of parameter 1 and β (the scaling parameter): $\nu_j \sim Beta(1, \beta)$. Hence, $1 - \nu_j$ is the probability that an item will be associated with a label with an index above j, given that it is not inferior to j. The probability that a true label t_i has value j is κ_j and follows the stick breaking construction:

$$\kappa_j = \nu_j \prod_{\tau=1}^{j-1} (1 - \nu_\tau).$$

That is, the true label of data item i, denoted t_i, is drawn from a multinomial distribution (similarly to the parametric model) of parameters with infinite dimension $\boldsymbol{\kappa}$. In the sequel, all equations are expressed directly with respect to $\boldsymbol{\nu}$.

Note that the confusion matrix $\boldsymbol{\pi}^{(k)}$ of each annotator k now also has an

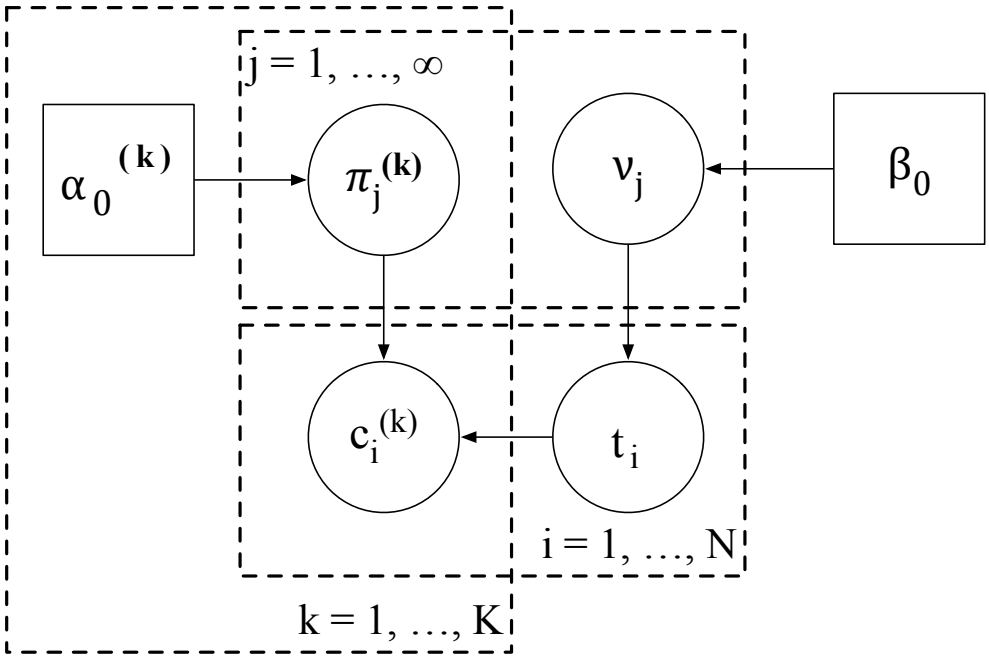

Figure 2: Nonparametric Bayesian annotator combination.

infinite number of rows. This is because $\boldsymbol{\pi}^{(\boldsymbol{k})}$ is conditioned over t_i, which is generated from a multinomial distribution of infinite dimension.

The joint probability distribution of the non-parametric *BAC* illustrated in Figure 2 can be expressed as follows:

$$p(\boldsymbol{\nu}, \boldsymbol{\Pi}, \boldsymbol{t}, \boldsymbol{c} | \boldsymbol{A_0}, \beta_0) = \prod_{i=1}^{N} [\prod_{j=1}^{\infty} \{\nu_j^{\mathbb{1}[t_i=j]} (1-\nu_j)^{\mathbb{1}[t_i<j]}\} \prod_{k=1}^{K} \pi^{(k)}_{t_i, c_i^{(k)}}] p(\boldsymbol{\nu}|\beta_0) p(\boldsymbol{\Pi}|\boldsymbol{A_0}). \tag{2}$$

Note that the index 0 over β_0 shows that it is our prior belief.

5. Variational Inference

Variational inference, contrary to *MCMC* methods such as Gibbs sampling [3], is known to be extremely fast [12] while still achieving a good

approximation of the joint probability distribution. This is important, especially in open-world classification scenarios where the number of true labels is not known and can be huge. In subsection 5.1, we first recall the intuition of the mean field variational Bayes. Then, in subsection 5.2, we present the specific equations for the NPBAC model. Finally, in subsection 5.3, we show how to handle the infinite dimension of the Dirichlet process as well as the variational algorithm.

5.1. Variational Method

This family of methods is used to approximate the computation of intractable integrals. In our particular context, this consists of finding a tractable distribution q which will approximate the true posterior distribution $p(Z|X)$, with Z being the parameters and X the observed data. This is performed using the Kullback-Leiber divergence (KL-divergence) [12] of q with respect to p, which is defined as follows:

$$\begin{aligned} D_{KL}(q||p) &= \int q(Z)\log\frac{q(Z)}{p(Z|X)}dZ \\ &= \int q(Z)\log\frac{q(Z)}{p(Z,X)}dZ + \log\, p(X) \\ &= \mathcal{L}(q) + \log\, p(X), \end{aligned} \tag{3}$$

where $\mathcal{L}(q)$ is called the variational free energy.
While formulating (3) as

$$\log\, p(X) = D_{KL}(q||p) - \mathcal{L}(q), \tag{4}$$

and since $p(X)$ is invariant regarding q, one has to minimize the variational free energy in order to minimize the KL-divergence.

Here we rely on the mean field theory [18] to estimate our model. One common shortcoming of this theory is that it assumes that the distribution $q(Z)$ factorizes among its latent variables and parameters as

$$q(Z) = \prod_{i=1}^{M} q_i(Z_i|X).$$

Using calculus of variations, it can be demonstrated that in order to minimize the variational free energy $\mathcal{L}(q)$, it is possible to work one variable at a time and that according to [12] the best form for q is:

$$\log\, q_i(Z_i|X) = \mathbb{E}_{j\neq i}[\log\, p(Z, X)] + \text{const.} \tag{5}$$

When developing this equation, choosing conjugated priors implies that $q_i(Z_i|X)$ will have the same distribution as $p(Z_i)$. On the contrary, if the prior is not conjugated, then $q_i(Z_i|X)$ will become intractable. Then circular dependencies will appear among the different variational equations. These dependencies actually describe the algorithm that will iterate until convergence. Convergence can be evaluated using the variational free energy $\mathcal{L}(q)$ [12, 23].

5.2. Variational Equations

Recall that under the mean field assumption, the latent variables of our model are assumed to factorize. Besides already independent variables, the variational factorization hypothesis of our problem is as follows:

$$q(\boldsymbol{t}, \boldsymbol{\Pi}, \boldsymbol{\nu}) = q(\boldsymbol{t})q(\boldsymbol{\Pi}, \boldsymbol{\nu}).$$

In the sequel, q is developed for each latent variable. We will show that $\boldsymbol{\Pi}$ and $\boldsymbol{\nu}$ can be derived separately.

The variational equation for the true labels variable $\boldsymbol{t}$ is

$$\ln\, q(\boldsymbol{t}) = \mathbb{E}_{\boldsymbol{\nu},\boldsymbol{\Pi}}[\ln\, p(\boldsymbol{\nu}, \boldsymbol{\Pi}, \boldsymbol{t}, \boldsymbol{c})] + \text{const.}$$

This equation can be formulated for each data item. For the sake of simplicity and to be in line with Simpson *et al.* [23], we use the following notation:

$$\begin{aligned} \ln\, \rho_{i,j} &= \mathbb{E}_{\boldsymbol{\nu},\boldsymbol{\Pi_j}}[\ln\, p(\boldsymbol{\nu}, \boldsymbol{\Pi_j}, t_i, c_i)] \\ &= \sum_{\tau=1}^{j-1}\{\mathbb{E}_{\nu_j}[\ln\, (1-\nu_\tau)]\} + \mathbb{E}_{\nu_j}[\ln\, \nu_j] + \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{\Pi_j}}[\ln\, \pi^{(k)}_{j,c_i^{(k)}}], \end{aligned} \tag{6}$$

yielding to the probability of a true label as

$$q(t_i = j) = \mathbb{E}_t[\mathbb{1}[t_i = j]] = \frac{\rho_{i,j}}{\sum_{\tau=1}^{\infty} \rho_{i,\tau}}. \tag{7}$$

where $\mathbb{1}[t_i = j]$ equals 1 if $t_i = j$ and 0 otherwise. For the sake of simplicity, let us consider the notations

$$\begin{aligned} N_j &= \sum_{i=1}^{N} \mathbb{E}_t[\mathbb{1}[t_i = j]] \text{ and} \\ N_j^+ &= \sum_{\tau=j+1}^{\infty} N_\tau, \end{aligned} \tag{8}$$

which respectively represent the number of data items that belong to class j and the number of items that belong to a class with an index above j. Let us also assume that the following notation represents the number of times an annotator k gave answer l when the true label was j:

$$N_{jl}^{(k)} = \sum_{i=1}^{N} \mathbb{1}[c_i^{(k)} = l]\mathbb{E}_t[\mathbb{1}[t_i = j]], \tag{9}$$

where $\mathbb{1}[c_i^{(k)} = l]$ equals 1 if $c_i^{(k)} = l$ and 0 otherwise.

The second term $q(\boldsymbol{\nu}, \boldsymbol{\Pi})$ can be developed as

$$q(\boldsymbol{\nu}, \boldsymbol{\Pi}) = \prod_{j=1}^{\infty} q(\boldsymbol{\nu_j}) \prod_{k=1}^{K} q(\boldsymbol{\pi_j^{(k)}}).$$

Indeed, from the joint distribution we note that $\boldsymbol{\nu}$ and $\boldsymbol{\Pi}$ can be factorized without any assumption.

As both variables can be evaluated separately, let us begin with ν_j:

$$\begin{aligned} \ln\, q(\nu_j) &= \sum_{i=1}^{N} \{\mathbb{E}_t[\ln\, (1-\nu_j)^{\mathbb{1}[t_i>j]} + \ln\, \nu_j^{\mathbb{1}[t_i=j]}]\} + \ln(1-\nu_j)^{\beta_0} + c_1 \\ &= N_j^+\, \ln\, (1-\nu_j) + N_j\, \ln\, \nu_j + (\beta_0 - 1)\ln\, (1-\nu_j) + c_1. \end{aligned} \tag{10}$$

It is possible to recognize the Beta distribution from the last equation:

$$q(\nu_j) \sim \text{Beta}(\nu_j|\gamma, \beta)$$

with

$$\gamma = 1 + N_j$$

$$\beta = \beta_0 + N_j^+.$$

Thus, the expectations over $\ln\, \nu_j$ and $\ln\, (1-\nu_j)$ are of the form:

$$\begin{aligned} \mathbb{E}[\ln\, \nu_j] &= \psi(\gamma) - \psi(\gamma+\beta) \\ \mathbb{E}[\ln\, (1-\nu_j)] &= \psi(\beta) - \psi(\gamma+\beta), \end{aligned} \tag{11}$$

with $\psi(x)$ being the digamma function. Like ν_j, the equations can be developed for all $q(\boldsymbol{\pi_j^{(k)}})$ (last term of $q(\boldsymbol{t}, \boldsymbol{\nu}, \boldsymbol{\Pi})$ and $q(\boldsymbol{\nu}, \boldsymbol{\Pi})$):

$$\begin{aligned} \ln\, q(\boldsymbol{\pi_j^{(k)}}) &= \sum_{i=1}^{N} \mathbb{E}_{t_i}[\mathbb{1}[t_i = j]] \ln\, \pi^{(k)}_{j,c_i^{(k)}} + p(\boldsymbol{\pi_j^{(k)}}|\boldsymbol{\alpha_{0,j}^{(k)}}) + c_2 \\ &= \sum_{l=1}^{L} \{N_{jl}^{(k)} \ln\, \pi_{jl}^{(k)} + (\alpha_{0,jl}^{(k)} - 1)\ln\, \pi_{jl}^{(k)}\} + c_2. \end{aligned} \tag{12}$$

It is possible to recognize the Dirichlet distribution from the previous equation:

$$q(\boldsymbol{\pi}_{\boldsymbol{j}}^{(\boldsymbol{k})}) \sim \text{Dirichlet}(\boldsymbol{\pi}_{\boldsymbol{j}}^{(\boldsymbol{k})} | \alpha_{j1}^{(k)}, ..., \alpha_{jL}^{(k)})$$

with

$$\alpha_{jl}^{(k)} = \alpha_{0,jl}^{(k)} + N_{jl}^{(k)},$$

yielding to the following expectation:

$$\mathbb{E}[\pi_{jl}^{(k)}] = \psi(\alpha_{jl}^{(k)}) - \psi(\sum_{m=1}^{L} \alpha_{jm}^{(k)}). \tag{13}$$

5.3. Variational Algorithm and Dirichlet Process

In this subsection, the variational algorithm is introduced based on the previous equations.

First, equations (7) and (8) have an infinite loop, while $\boldsymbol{\nu}$ and $\boldsymbol{\Pi}$ have infinite dimensions. This is directly related to the Dirichlet process and it makes their computations impossible, but fortunately this is neither wanted nor required. Indeed, given a dataset of finite size, it is obvious that there cannot be more labels than data items if each one is associated with a single label. In other scenarios, the maximum number of possible labels can also be the number of different annotator outputs. Like Blei and Jordan [4], let us set an upper bound to the total number of possible labels. Let T denote this upper bound, thus giving the following equations:

$$\nu_T = 1$$

$$\kappa_T = \prod_{\tau=1}^{T-1} (1 - \nu_\tau),$$

which means that there is a probability of 1 that the label cannot have an index superior to T.

Algorithm 1 describes the procedure developed from the circular dependencies appearing in the previous equations. At line 1, all variables of the model are initialized either randomly or with some prior knowledge. Random initialization with a uniform prior requires that all annotators start with the same or with a similar random confusion matrix. In a totally random initialization with all annotators having a different random confusion matrix with a uniform prior, the uniform distribution would arise after convergence, thus leading to incorrect classification results.

The number of iterations set in Algorithm 1 can be either a fixed parameter or directly evaluated based on the variational method. We do not go into further detail about the evaluation of the convergence in this paper, but they are given in [23].

Algorithm 1: Running the variational Bayes algorithm

input : $\{c_i^{(k)}\}_{i\in 1,\dots,N, k\in 1,\dots,K}$, β

1 initialization;

2 **for** *iteration* **in** *nb_iteration* **do**

3 estimate ln $q(\boldsymbol{t})$ (Equation 7);

4 estimate ln $q(\boldsymbol{\nu})$ (Equation 11);

5 estimate ln $q(\boldsymbol{\pi})$ (Equation 13);

6. Prior over the Concentration Parameter

Although setting the correct value for the concentration parameter β is less restrictive than setting the number of classes J, setting a prior over its values to describe its uncertainty $p(\beta|\theta)$ might be beneficial. Figure 3 presents the Bayesian graphical model when the concentration parameter β follows a distribution of parameter θ. Unfortunately, ν_j is drawn from a Beta distribution (of parameter $[1, \beta]$), whose own conjugated prior is known to be not in a closed form. Recall that conjugacy is a fundamental property required to obtain a tractable variational approximation of the joint distribution. As an alternative, Subsection 6.1 describes how to approximate the variational equation for β using the Laplace method. Then Section 6.2 shows the updated variational algorithm taking into account the β equation.

6.1. Approximating the Variational Equation for β

In order to approximate the variational equation for parameter β, we use the Laplace method, which proposes a solution to evaluate integrals of the form

$$\int_a^b e^{Mf(x)} dx,$$

where f is a twice differentiable function, while $M \in \mathbb{R}$ and both a, b can be infinite, such as used in [27].

Several assumptions are necessary:

1. β must be real valued. This is already the case.
2. $p(\beta|\theta)$ must be twice differentiable with respect to β. This is important as in the following we do not restrict β to a specific distribution.

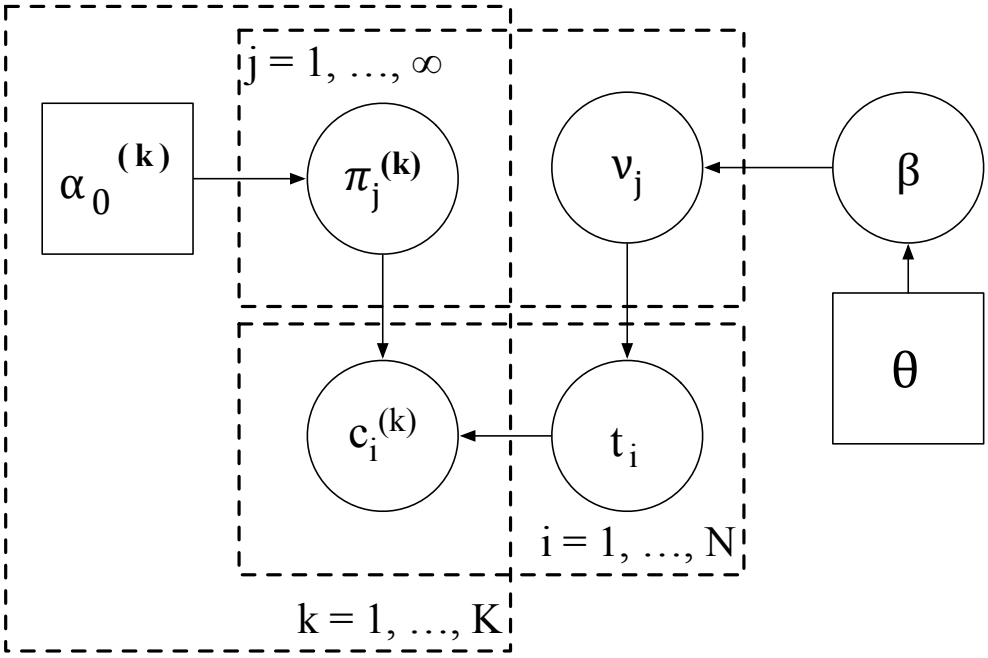

Figure 3: Non-parametric Bayesian annotator combination with a prior over the concentration parameter.

3. $p(\nu_j|\boldsymbol{\phi})$ with $\boldsymbol{\phi} = [1, \beta]$ must be in the exponential family. This is the case as ν_j follows a Beta distribution, which can therefore be expressed as follows [1]:

$$p(\nu_j|\boldsymbol{\phi}) = h(\nu_j)\exp\{\eta(\boldsymbol{\phi})^T t(\nu_j) - a(\eta(\boldsymbol{\phi}))\},$$

where $h(\nu_j)$ is a function called *base measure*, $t(\nu_j)$ is the *sufficient statistic*, $\eta(\boldsymbol{\phi})$ is the natural parameter and $a(\eta(\boldsymbol{\phi}))$ is the log partition function. $\eta(\boldsymbol{\phi})$ is assumed to be twice differentiable: this is the case for the Beta distribution. In our context, as ν_j is drawn from a Beta distribution, these elements take the following values:

- $h(\nu_j) = 1$,

- $t(\nu_j) = [\ln\, \nu_j, \ln\, (1-\nu_j)]^T$,
- $\eta(\boldsymbol{\phi}) = [\phi_1, \phi_2]^T$ where $\boldsymbol{\phi} = [1, \beta]$,
- $a(\eta(\boldsymbol{\phi})) = \ln\, \Gamma(\phi_1) + \ln\, \Gamma(\phi_2) - \ln\, \Gamma(\phi_1 + \phi_2)$.

Let us estimate $q(\boldsymbol{\phi})$ using the Laplace method as

$$q(\boldsymbol{\phi}) \propto \exp\{\sum_j [\eta(\boldsymbol{\phi})^T \mathbb{E}_\nu[t(\nu_j)] - a(\eta(\boldsymbol{\phi}))] + \ln\, p(\phi_2|\theta)\} = \exp\{f(\boldsymbol{\phi})\}$$

$$f(\boldsymbol{\phi}) \triangleq \sum_j [\eta(\boldsymbol{\phi})^T \mathbb{E}_\nu[t(\nu_j)] - a(\eta(\boldsymbol{\phi}))] + \ln\, p(\phi_2|\theta).$$

Function f can be approximated using the second-order Taylor approximation around $\boldsymbol{\phi}^\star$ which denotes the value that maximizes the function, such that

$$f(\boldsymbol{\phi}) \approx f(\boldsymbol{\phi}^\star) + \nabla f(\boldsymbol{\phi}^\star)(\boldsymbol{\phi} - \boldsymbol{\phi}^\star) + \frac{1}{2}(\boldsymbol{\phi} - \boldsymbol{\phi}^\star)^T \nabla^2 f(\boldsymbol{\phi}^\star)(\boldsymbol{\phi} - \boldsymbol{\phi}^\star).$$

Note that since $\boldsymbol{\phi}^\star$ maximizes f, then $\nabla f(\boldsymbol{\phi}^\star) = 0$, thus leading to

$$q(\boldsymbol{\phi}) \propto \ \exp\{f(\boldsymbol{\phi}^\star) + \frac{1}{2}(\boldsymbol{\phi} - \boldsymbol{\phi}^\star)^T \nabla^2 f(\boldsymbol{\phi}^\star)(\boldsymbol{\phi} - \boldsymbol{\phi}^\star)\}$$

$$q(\boldsymbol{\phi}) \approx \mathcal{N}(\boldsymbol{\phi}^\star, -\nabla^2 f(\boldsymbol{\phi}^\star)^{-1}). \tag{14}$$

The Gaussian form of our approximation results from the Taylor series. The vector $\boldsymbol{\phi}^\star$ still needs to be computed. This is done by formulating f as

$$\begin{aligned} f(\boldsymbol{\phi}) =& \sum_{j=1}^{T} \{\phi_1 \mathbb{E}_\nu[\ln(\nu_j)] + \phi_2 \mathbb{E}_\nu[\ln(1-\nu_j)] - \\ & (\ln\, \Gamma(\phi_1) + \ln\, \Gamma(\phi_2) - \ln\, \Gamma(\phi_1 + \phi_2))\} + \ln\, p(\phi_2|\theta) \\ =& \sum_{j=1}^{T} \{\mathbb{E}_\nu[\ln(\nu_j)] + \beta\ \mathbb{E}_\nu[\ln(1-\nu_j)] - \\ & (\ln\, \Gamma(1) + \ln\, \Gamma(\beta) - \ln\, \Gamma(1+\beta))\} + \ln\, p(\beta|\theta), \end{aligned} \tag{15}$$

where $\mathbb{E}_\nu[\ln(1-\nu_j)]$ and $\mathbb{E}_\nu[\ln(\nu_j)]$ are updated from Equation (11). A numerical method (*e.g.* Gradient Descent) can be used to find $\boldsymbol{\phi}^\star = [1, \beta^\star]$.

6.2. The Variational Algorithm

Algorithm 2 presents the modified version of Algorithm 1, while taking the estimation of β into account. In particular, note that at line 5 the update procedure for β is not as simple as for the other variables and requires maximization.

Algorithm 2: Running the variational Bayes algorithm

input : $\{c_i^{(k)}\}_{i\in 1,\dots,N, k\in 1,\dots,K}$, θ and $p(\beta|\theta)$

1 initialization;
2 **for** *iteration* **in** *nb_iteration* **do**
3 estimate $\ln q(\boldsymbol{t})$ (Equation 7);
4 estimate $\ln q(\boldsymbol{\nu})$ (Equation 11);
5 estimate $\ln q(\beta)$ (Equation 15, through maximization);
6 estimate $\ln q(\boldsymbol{\pi})$ (Equation 13);

7. Experiments

In this section, the experimental evaluation of NPBAC is presented. First, Subsection 7.1 describes the setup of our experiments. Then Subsection 7.2 presents and discusses the results.

7.1. Setup

To evaluate the NPBAC model, we compare it to the classical BAC model that we built on [15, 23]. We do not evaluate other published Bayesian

methods as they could also have benefited from a non-parametric distribution over the possible true labels.

7.1.1. Datasets

Three datasets are used in our experiments. The first one is related to dogs classification introduced in [29]. It contains images of four dog breeds taken from the Stanford dogs dataset [9]. The classification task was achieved through Amazon Mechanical Turk [6]. The two other datasets come from Pl@ntnet [13], an innovative participatory sensing platform that relies on image-based plant identification as a means to enlist non-expert annotators and facilitate the production of botanical observation data. Pl@ntnet relies on a mobile application available on iOS and Android, which enables users to take images of plants and in return receive the most likely species. The data stream generated through the mobile application consists of a set of plant observations. These data can be used to monitor biodiversity, invasive species and general plant population structure. However, although machine learning might successfully identify some observations, the data stream is highly noisy and therefore needs to be human validated.

With the aim of producing a clean dataset of French flora, the authors have developed a crowdsourcing platform called The Plant Game [21][3]. The Plant Game addresses three issues. First, it offers a training module enabling users to gain expertise and then become capable of recognizing species they have been trained on. All users are trained on different plant species so that they are complementary (on more or fewer species depending on their initial

[3] `http://theplantgame.com`

expertise and learning capacity). Second, the Plant Game assigns plant observations to users that are likely to identify them. Finally, identification propositions are aggregated using a Bayesian model developed from that proposed by Simpson *et al.* [23]. Part of The Plant Game dataset is derived from the LifeClef challenge [7] and is therefore associated with a true label.

In summary, the main characteristics of the three datasets we use in our experiments are:

- **Dog Dataset**: which contains 807 pictures of dogs belonging to 1 of 4 possible species, and 109 different annotators. There is a bijection between the annotators' propositions and the true labels.

- **Plant Dataset 1**: which consists of all observations from the 5 most popular species/labels in *The Plant Game*. This subset contains 400 annotators and 155 items. For these items the annotators proposed 93 different labels, from which only 5 were correct.

- **Plant Dataset 2**: which consists of all observations from the 3 most popular species/labels and all those from the 2 most confused ones by the annotators in *The Plant Game*. Confusion here means that given these two species A and B, A is often taken for B and reciprocally. This subset contains 677 annotators and 200 items for which the annotators proposed 95 different labels from which only 5 are correct. The goal of this dataset is to determine if the non-parametric model is able to model more complex confusion by dividing some classes into two sub-classes, thus achieving a better accuracy.

7.1.2. Experimental Scenarios

We study three different scenarios in which we could apply the NPBAC model:

1. **Closed-world classification:** we use the *Dog Dataset* [29] where the votes are a bijection with true labels,
2. **Open-world classification:** we use the two *Plant Datasets* [13, 21] as the annotators are not obliged to answer within the set of true classes of which they are not necessarily even aware,
3. **Novelty detection:** we use *Plant Dataset 2* from The Plant Game. In all experiments, we randomly remove 1 true class from the training set and evaluate the different models on the test set (see below).

7.1.3. Protocol and Metrics

All scenarios are tested using a k-fold cross-validation method, assessing various training and test set ratios. In addition, the first two scenarios are also tested in an unsupervised manner (the training set has size 0).

We measure the resulting classification accuracy for each model. Since those Bayesian models can divide the datasets into a large number of classes – up to one label per data item –, we measure the resulting number of predicted labels (at least one item associated with it). This avoids over-fitting confirmation bias (e.g. splitting the dataset in too many classes). Each experiment is run ten times and for each metric we show the mean, as well as its standard deviation.

7.1.4. Models Initialization and Parameters Choices

All models are randomly initialized. The annotators' abilities are directly derived from the data (possibly from the training set, if available). In particular, we consider each $\boldsymbol{\alpha}^{(k)}$ to be a matrix of 1.0, and $\boldsymbol{\nu}$ to be a vector of 1.0 in the baseline parametric model (BAC). Then we sample a matrix following a Dirichlet distribution of parameters $\boldsymbol{\alpha}^{(k)}$ and assign the same matrix for all annotators. Moreover, in the baseline parametric model, $\boldsymbol{\kappa}$ is drawn from a Dirichlet distribution of parameter $\boldsymbol{\nu}$, while in the non-parametric model all ν_j are drawn from a Beta distribution of parameters 1 and β. We test different β values in the experiments while setting a Gamma prior over it. We choose a Gamma distribution with a very large standard deviation of 120 and a mean of 240. This corresponds to the previous manually set values of β.

To set J in the parametric model, we consider two possible solutions. First, the number of true labels is known and is the J value. Second, it is unknown and we choose the number of different labels proposed by the annotators.

To set the upper bound T, we choose the number of different labels for the plant datasets and 10 for the dog dataset to give the model some flexibility.

7.2. Results

7.2.1. Closed-World Scenario

Table 1 presents an unsupervised classification on the dog dataset, which consists of a closed-world scenario. On this dataset, the majority voting obtains a precision of 0.82, as presented in [29]. In addition, while the parametric BAC tends to underestimate the true number of labels, the non-

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=4)	0.796 (0.03)	3.7 (0.45)
Non-parametric BAC ($\beta = 1$)	0.83 (0.07)	4.6 (0.72)
Non-parametric BAC ($\beta = 10$)	**0.857** (0.03)	5.9 (1.76)
Non-parametric BAC ($\beta = 100$)	0.826 (0.05)	6.2 (1)
Non-parametric BAC (prior Gamma)	0.81 (0.07)	4.5 (0.5)

Table 1: Dog dataset: votes over 4 possible labels, 4 true labels, sampling (train/test): 0%/100%.

parametric model, whatever the β value, tends to overestimate it. However, the non-parametric model obtains consistently better accuracy compared to the classical parametric model. In addition, setting a Gamma prior over the concentration parameter β results in the worst accuracy for the non-parametric model, even though it is still better than the parametric model. Note however that both the parametric BAC with $J = 4$ and the Non-parametric BAC with a gamma prior are both slightly outperformed by the majority voting in this experiment.

Table 2 presents the results when the models are trained on 10% of the dataset. Here all methods perfectly estimate the number of true labels and the accuracy is almost always higher than in the unsupervised experiment. The parametric and non-parametric models achieve similar accuracy. We also tried with a training set of 50% and the results were identical – not shown here for the sake of clarity. Note that both the parametric approaches and non-parametric approaches largely outperform the majority voting approach (0.8209) when training data are available.

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=4)	0.842 (0.006)	4 (0)
Non-parametric BAC ($\beta = 1$)	0.837 (0.004)	4 (0)
Non-parametric BAC ($\beta = 10$)	0.84 (0.001)	4 (0)
Non-parametric BAC ($\beta = 100$)	0.839 (0.005)	4 (0)
Non-parametric BAC (prior Gamma)	0.841 (0.007)	4 (0)

Table 2: Dog dataset: votes over 4 possible labels, 4 true labels, sampling (train/test): 10%/90%.

In this context of closed-world scenarios, the use of NPBAC over the parametric model is beneficial since the performances are improved or at least stable, whatever the configuration tested.

7.2.2. Open-World Scenario

Tables 3 and 4 present the unsupervised classification in an open-world scenario on the two datasets related to plant classification. In the following experiments, we do not compare with the majority voting as there is no bijection between the annotators' outputs and the actual expected classes. However, a detailed comparison between BAC and majority voting is available in our previous work [21]. Again, the parametric model (even when we increase the number of possible labels) tends to underestimate the number of true labels. This is slightly unexpected as we may intuitively assume that the higher the number of possible true labels in the Dirichlet distribution, the higher the probability of completely over-fitting the data. Note however that Rasmussen and Gharhamani have shown that Bayesian models tend

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=5)	0.86 (0.1)	3.5 (0.8)
Baseline parametric BAC (J=93)	0.934 (0.17)	4.3 (0.78)
Non-parametric BAC ($\beta = 10$)	0.98 (0.02)	5.4 (0.48)
Non-parametric BAC ($\beta = 100$)	0.95 (0.12)	5.4 (1)
Non-parametric BAC ($\beta = 500$)	0.95 (0.08)	5.7 (0.46)
Non-parametric BAC ($\beta = 1,000$)	0.88 (0.13)	5 (1)
Non-parametric BAC (prior Gamma)	0.96 (0.03)	5.4 (0.91)

Table 3: Plant dataset 1: votes over 93 possible labels, 5 true labels, sampling (train/test): 0%/100%.

to not overfit the data [19]. Conversely, the non-parametric model slightly overestimates the number of labels but obtains a better approximation of this number and a better overall accuracy. For instance, when $\beta = 10$, the accuracy is 12 points higher than the classical BAC model. Note also that when β tends to get big values (*e.g.* $1,000$), the accuracy drops. This can be explained by the upper bound T on the dimension of the Dirichlet process. Indeed, when β is big, the items tend to easily obtain classes with a high index j. Unfortunately, some items end up in class $j = T$, where $\nu_T = 1$. This means that there is a probability of 1.0 that if these items are not in a class with index $j < T$, they can only be in a class with index T and get stuck within it. Note in Tables 3 and 4 that when the concentration parameter β follows a Gamma prior, the model achieve a very good accuracy as well as a good approximation of the number of true labels.

Tables 5 and 6 present the results when the models have access to a

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=5)	0.74 (0.11)	4.1 (0.83)
Baseline parametric BAC (J=95)	0.808 (0.14)	5.1 (0.93)
Non-parametric BAC ($\beta = 500$)	0.91 (0.07)	6.0 (1.0)
Non-parametric BAC ($\beta = 1,000$)	0.86 (0.1)	6.8 (1.33)
Non-parametric BAC (prior Gamma)	0.84 (0.12)	5 (1.1)

Table 4: Plants dataset 2: votes over 95 possible labels, 5 true labels, sampling (train/test): 0%/100%.

training set of 10%. Again, the non-parametric model is better at estimating the true number of labels. In addition, even though the training set contains 5 classes, the non-parametric model is still capable of creating new classes from the test set. Finally, all methods now tend to have good accuracy.

Tables 7 and 8 present the results when using a training set of 50%. The parametric model as well as the non-parametric model (when $\beta = 10$ and when it follows a gamma prior) correctly estimate the real number of true labels. When β has a bigger value, the model generally considers an additional class but achieves better accuracy. In addition, the non-parametric model is still better in terms of accuracy than the parametric model. For instance, in Table 8, when $\beta \in \{500, 1000\}$, the non-parametric model estimates that there are six types of confusion in the data (and therefore of labels) and achieves 100% accuracy.

To sum up, these experiments show that the NPBAC model has stronger abilities in open-world contexts, most notably when the classes are "confused".

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=5)	0.96 (0.03)	4.6 (0.49)
Baseline parametric BAC (J=93)	0.96 (0.03)	4.7 (0.5)
Non-parametric BAC ($\beta = 10$)	0.97 (0.03)	4.7 (0.45)
Non-parametric BAC ($\beta = 100$)	0.98 (0.03)	5.2 (0.6)
Non-parametric BAC ($\beta = 500$)	0.93 (0.06)	5.3 (0.46)
Non-parametric BAC ($\beta = 1,000$)	0.93 (0.07)	5.4 (0.48)
Non-parametric BAC (prior Gamma)	0.97 (0.08)	4.9 (0.46)

Table 5: Plant dataset 1: votes over 93 possible labels, 5 true labels, sampling (train/test): 10%/90%.

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=5)	0.95 (0.05)	4.6 (0.49)
Baseline parametric BAC (J=95)	0.95 (0.05)	4.7 (0.5)
Non-parametric BAC ($\beta = 10$)	0.97 (0.04)	4.7 (0.46)
Non-parametric BAC ($\beta = 1,000$)	0.91 (0.07)	5.5 (0.5)
Non-parametric BAC (prior Gamma)	0.95 (0.05)	4.6 (0.48)

Table 6: Plant dataset 2: votes over 95 possible labels, 5 true labels, sampling (train/test): 10%/90%.

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=5)	0.993 (0.04)	5 (0)
Baseline parametric BAC (J=93)	0.96 (0.03)	5 (0)
Non-parametric BAC ($\beta = 10$)	0.994 (0.03)	5 (0)
Non-parametric BAC ($\beta = 100$)	1.0 (0)	5.1 (0.3)
Non-parametric BAC ($\beta = 500$)	0.996 (0.04)	6 (0)
Non-parametric BAC ($\beta = 1,000$)	0.995 (0.03)	6 (0)
Non-parametric BAC (prior Gamma)	0.995 (0.04)	5 (0)

Table 7: Plant dataset 1: votes over 93 possible labels, 5 true labels, sampling (train/test): 50%/50%.

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=5)	0.994 (0.005)	5 (0)
Baseline parametric BAC (J=95)	0.994 (0.006)	5 (0)
Non-parametric BAC ($\beta = 100$)	1.0 (0)	5.4 (0.49)
Non-parametric BAC ($\beta = 500$)	1.0 (0)	6 (0)
Non-parametric BAC ($\beta = 1,000$)	1.0 (0)	6 (0)
Non-parametric BAC (prior Gamma)	0.999 (0.003)	5 (0)

Table 8: Plant dataset 2: votes over 95 possible labels, 5 true labels, sampling (train/test): 50%/50%.

Model	accuracy μ (σ)	nb labels μ (σ)
Baseline parametric BAC (J=5)	0.89 (0.05)	4.2 (0.44)
Baseline parametric BAC (J=95)	0.904 (0.05)	4.2 (0.45)
Non-parametric BAC ($\beta = 100$)	0.98 (0.03)	4.8 (0.4)
Non-parametric BAC ($\beta = 500$)	0.96 (0.03)	5 (0)
Non-parametric BAC ($\beta = 1,000$)	0.95 (0.04)	5 (0)
Non-parametric BAC (prior Gamma)	0.98 (0.03)	5 (0)

Table 9: Plant dataset 1: votes over 93 possible labels, 5 true labels, sampling (train/test): 10%/90% and one missing class in the training set.

7.2.3. Novelty Detection Scenario

In this last experiment presented in Table 9 and related to novelty detection, the training set only contains 4 classes while 5 are available in the test set. Surprisingly, the parametric model is sometimes capable of estimating over four labels. However, the non-parametric model is more flexible and thus more capable of estimating the true number of labels. As expected, since the non-parametric model is capable of discovering all labels most of the time, it also achieves better accuracy than the parametric model. Moreover, we can observe that setting a Gamma prior enables us to obtain better accuracy even though less prior knowledge is given to the model.

Here again, NPBAC outperformed the baseline parametric BAC when considering novelty detection.

8. Conclusion

In this paper, we have presented the NPBAC model to address the problem of combining annotators and learning their confusion when the number of possible "true labels" is not known or when some of the classes are highly confused. Most notably, we have proposed the following original contributions.

We show how the non-parametric model is related to the parametric model through the Dirichlet process. We also show how to express the joint probability of the NPBAC model using the "stick-breaking" construction of the Dirichlet process.

We show how the variational equations of the non-parametric model can be derived and present the variational algorithm to infer the joint probability of the model given the observed data. In addition, we show how to set a prior distribution over the Dirichlet process and how to infer its variational equation using the Laplace method.

Overall, we compare the non-parametric model to the parametric model on three datasets and three different scenarios in unsupervised, semi-supervised and supervised classification. The first scenario is the typical closed-world scenario where all labels are known and where the votes and labels form a bijection. The second scenario is related to an open-world classification scenario where the labels are no longer known and the votes no longer form a bijection with the true labels. Finally, the last scenario is related to novelty detection. Here the models are trained on a dataset that does not contain all labels from the test set. In summary, our experiments show that in these three scenarios the non-parametric model correctly approximates the number

of true labels (even more than the parametric model) and largely outperforms the parametric model in terms of accuracy, especially when no or few training data are available. Both models tend to converge to the same accuracy when the training set is large (*i.e.* 50% of the total dataset).

Future studies could address the problem of scalability of the non-parametric Bayesian annotator combination model. In particular, the equations model all possible confusions that might exist given an annotator but only a subset of those may actually appear.

9. References

[1] Andersen, E. B. (1970). Sufficiency and exponential families for discrete sample spaces. *Journal of the American Statistical Association*, 65(331):1248–1255.

[2] Bendale, A. and Boult, T. (2015). Towards open world recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1893–1902.

[3] Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer.

[4] Blei, D. M. and Jordan, M. I. (2006). Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143.

[5] Bragg, J., Weld, D. S., et al. (2016). Optimal testing for crowd workers. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 966–974.

[6] Buhrmester, M., Kwang, T., and Gosling, S. D. (2011). Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5.

[7] Champ, J., Lorieul, T., Servajean, M., and Joly, A. (2015). A comparative study of fine-grained classification methods in the context of the lifeclef plant identification challenge 2015. In *CLEF: Conference and Labs of the Evaluation forum*, volume 1391.

[8] Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28.

[9] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255.

[10] Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230.

[11] Fortson, L., Masters, K., Nichol, R., Edmondson, E., Lintott, C., Raddick, J., and Wallin, J. (2012). Galaxy zoo. *Advances in machine learning and data mining for astronomy*, 2012:213–236.

[12] Fox, C. W. and Roberts, S. J. A tutorial on variational bayesian inference. *Artificial intelligence review*, 38(2):85–95.

[13] Joly, A., Bonnet, P., Goëau, H., Barbe, J., Selmi, S., Champ, J., Dufour-Kowalski, S., Affouard, A., Carré, J., Molino, J.-F., et al. (2016). A look inside the pl@ntnet experience. *Multimedia Systems*, 22(6):751–766.

[14] Kamar, E., Hacker, S., and Horvitz, E. (2012). Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 467–474.

[15] Kim, H.-C. and Ghahramani, Z. (2012). Bayesian classifier combination. In *International conference on artificial intelligence and statistics*, pages 619–627.

[16] Littlestone, N. and Warmuth, M. K. (1989). The weighted majority algorithm. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 256–261.

[17] Moreno, P. G., Artes-Rodriguez, A., Teh, Y. W., and Perez-Cruz, F. (2015). Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research*, 16:1607–1627.

[18] Parisi, G. (1988). *Statistical field theory*. Addison-Wesley.

[19] Rasmussen, C. E. and Ghahramani, Z. (2001). Occam's razor. *Advances in neural information processing systems*, pages 294–300.

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

[21] Servajean, M., Joly, A., Shasha, D., Champ, J., and Pacitti, E. (2017). Crowdsourcing thousands of specialized labels: A bayesian active training approach. *IEEE Transactions on Multimedia*, 19(6):1376–1391.

[22] Sethuraman, J. (1994). A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650.

[23] Simpson, E., Roberts, S. J., Psorakis, I., and Smith, A. (2013). Dynamic bayesian combination of multiple imperfect classifiers. *Decision making and imperfection*, 474:1–35.

[24] Simpson, R., Page, K. R., and De Roure, D. (2014). Zooniverse: observing the world's largest citizen science platform. In *Proceedings of the 23rd international conference on world wide web*, pages 1049–1054.

[25] Tulyakov, S., Jaeger, S., Govindaraju, V., and Doermann, D. (2008). Review of classifier combination methods. *Machine Learning in Document Analysis and Recognition*, pages 361–386.

[26] Venanzi, M., Guiver, J., Kazai, G., Kohli, P., and Shokouhi, M. (2014). Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on World wide web*, pages 155–164.

[27] Wang, C. and Blei, D. M. (2013). Variational inference in nonconjugate models. *Journal of Machine Learning Research*, 14(Apr):1005–1031.

[28] Welinder, P. and Perona, P. (2010). Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *Computer Vision and Pattern*

Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pages 25–32.

[29] Zhou, D., Basu, S., Mao, Y., and Platt, J. C. (2012). Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203.