# Development of a Variational Bayes algorithm for Bayesian semiparametric novelty detection

## Bayesian Statistics Project

Alessandra Guglielmi
Mario Beraha

Academic Year 2021/2022
Mathematical Engineering, Politecnico di Milano

Luca Benedetti, Eric Boniardi, Leonardo Chiani,
Jacopo Ghirri, Fabio Lavezzo, Marta Mastropietro

Tutors: Francesco Denti, Andrea Cappozzo

15 February 2022

# Contents

# 1 Introduction

Novelty detection consists in classifying observations in a clustering fashion, while only having information about a portion of the total number of sub-populations from which the observations are actually drawn.
In practice, this means having a labeled training set and an unlabeled test set where it is possible to encounter novelties, *i.e.*, observations originated from populations not seen in the training set.
In Denti et al. [2021] a two stage procedure to approach this problem is proposed: in the first stage the characteristics of the classes observed in the training set are extracted; in the second stage a Bayesian semiparametric mixture of the known groups plus a novelty term is fitted, where the latter flexibly accounts for all the previously unseen classes.

From the application point of view, one could exploit traditional MCMC methods, but they result to be quite inefficient, if not even unfeasible over large multidimensional datasets.
Our contribution is to solve the scalability issues of MCMC by applying a variational inference approach, improving the computational efficiency of the method.

## 1.1 BRAND: a two-stage model for novelty detection

We first describe the two-stage model proposed in Denti et al. [2021].
The first stage is carried out exploiting a fully labeled dataset, from which we extract robust minimum regularized covariance determinant (MRCD) estimates for the location and scale parameters of the observed populations.
The second stage consists in building a model as a combination of two Mixture models:

- A *Multivariate Gaussian* Mixture with conjugate *Normal-inverse-Wishart* prior distributions for modeling the populations observed in the training set with a fixed and known number of components, and informed prior obtained through the estimates computed at the first stage.
- A *Gaussian Dirichlet Process* Mixture built on top of another *Normal-inverse-Wishart* distribution for novelty identification, allowing an unknown number of components.

The proposed model then reads as follows:

$$
\begin{aligned}
&\boldsymbol{y}_m|\widetilde{\boldsymbol{\Theta}}, \xi_m \overset{ind}{\sim} N(\widetilde{\boldsymbol{\Theta}}_{\xi_m}) && m = 1, \ \dots, M,\\
&\xi_m|\widetilde{\boldsymbol{\pi}} \overset{iid}{\sim} \sum_{k=1}^{\infty} \widetilde{\pi}_k \cdot \delta_k(\cdot) && m = 1, \ \dots, M,\\
&\widetilde{\pi}_k = {\pi_k}^{\mathbb{1}\{0<k\leq J\}} \cdot (\pi_0 \cdot \omega_{k-J})^{\mathbb{1}\{k\geq J\}} && k \in \mathbb{N},\\
&\widetilde{\boldsymbol{\Theta}}_k = (\boldsymbol{\Theta}_k^{obs})^{\mathbb{1}\{0<k\leq J\}} \cdot (\boldsymbol{\Theta}_k^{nov})^{\mathbb{1}\{k\geq J\}} && k \in \mathbb{N},\\
&\boldsymbol{\pi} \sim \mathit{Dirichlet}(\alpha_0, \alpha_1, ..., \alpha_J),\\
&\boldsymbol{\omega} \sim \mathit{SB}(\gamma),\\
&\boldsymbol{\Theta}_k^{obs} \overset{ind}{\sim} \mathit{NIW}(\boldsymbol{\mu}_k^{\star obs}, \nu_k^{\star obs}, \lambda_k^{\star obs}, \boldsymbol{\Psi}_k^{\star obs}) && k = 1, \ \dots, J,\\
&\boldsymbol{\Theta}_k^{nov} \overset{iid}{\sim} \mathit{NIW}(\boldsymbol{\mu}_0^{nov}, \nu_0^{nov}, \lambda_0^{nov}, \boldsymbol{\Psi}_0^{nov}) && k = J+1, \ \dots, \infty,
\end{aligned}
$$

where $SB(\gamma)$ denotes a *Stick Breaking* process with parameter $\gamma$ and $\{(\boldsymbol{\mu}_k^{\star obs}, \nu_k^{\star obs}, \lambda_k^{\star obs}, \boldsymbol{\Psi}_k^{\star obs})\}_{k=1}^{J}$ are defined according to the robust estimates obtained in the first stage.
This model, developed in Denti et al. [2021], takes the name of *Bayesian Robust Adaptive model for Novelty Detection* (BRAND), and is the core element for the development of this project.

## 1.2 Theoretical foundation of Variational Inference

The variational inference approach consists in rephrasing a statistical simulation problem into an optimization one, as stated in Blei et al. [2017].
The posterior distribution is to be approximated, via an optimization algorithm, by a different distribution whose properties can be a priori defined.
The specific application of this approach on BRAND exploits Mean-Field Variational inference. This

means that the variational family, i.e., the family of distribution among which we want to find the best approximation of the posterior, is the family where variables are mutually independent and each one is governed by a distinct factor.
So, practically speaking, we approximate the posterior distribution $p$ with a distribution $q \in \mathbb{Q}$ defined as:

$$q(\boldsymbol{z}) = \prod_j q_j(z_j).$$

Then, as in any optimization framework, we need to define an objective function. In this setting the chosen loss function is the *Kullback-Leibler* (KL) divergence between the true posterior distribution and the variational approximation.
It should be noted that it is standard practice in this setting to rephrase the KL divergence into the *Evidence Lower Bound* (ELBO). This quantity can be treated as an equivalent loss function, but it allows for the computation and evaluation of the posterior up to a normalization constant. In fact we have:

$$D_{KL}(P||Q) = \mathbb{E}_q[\log q(\boldsymbol{z})] - \mathbb{E}_q[\log p(\boldsymbol{z}, \boldsymbol{y})] + \log p(y).$$

Note that all expectations are computed with respect to $q$, the variational approximation of the posterior. This implies that the marginal log probability $\log p(y)$ does not depend on the variational parameters and can be treated as a constant during the optimization procedure, in other words it can be ignored during computations. It is hence defined:

$$ELBO(q) = \mathbb{E}_q[\log p(\boldsymbol{z}, \boldsymbol{y})] - \mathbb{E}_q[\log q(\boldsymbol{z})]$$

which will need to be maximized in order to minimize the KL divergence.
Once we defined a loss function, we need an optimization algorithm.
The most used in this setting is Coordinate Ascent Variational Inference (CAVI), which consists in a one-variable-at-a-time optimization procedure, exploiting the independence seeded in the variational family. The pseudo-code for the algorithm is reported in Algorithm 1.

**Algorithm 1:** Coordinate ascent variational inference(CAVI)

**Input:** *A model* $p(\boldsymbol{y}, \boldsymbol{z})$, *a data set* $\boldsymbol{y}$
**Output:** *A variational density* $q(\boldsymbol{z}) = \prod_{j=1}^{m} q_j(z_j)$
**Initialize:** $Variational\ factors\ q_j(z_j)$
**while** *ELBO has not converged* **do**
  **for** $j \in \{1, ..., m\}$ **do**
    Set $q_j(z_j) \propto \exp\left(\mathbb{E}_{-j}[\log(p(z_j|\boldsymbol{z}_{-j}, \boldsymbol{y})]\right)$
  **end**
  *Compute* $ELBO(q) = \mathbb{E}[\log(p(\boldsymbol{z}, \boldsymbol{y}))] - \mathbb{E}[\log(q(\boldsymbol{z}))]$
**end**
**return** $q(\boldsymbol{z})$

Notice that at each step of the algorithm there is the need to compute the expected value of the log density of each full conditional with respect to the variational distribution of all other parameters and variables. Such computations need to be derived in closed form. Notice also that this method is liable to local minima problem. A possible ways to overcome that is to correctly initialize the variational factors.
Overall, the output of the method can be summarized as:

$$q^*(\boldsymbol{z}) = \underset{q(\boldsymbol{z}) \in Q}{\arg\min} D_{KL}\left(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{y})\right)$$

where $p$ is the posterior distribution, $\boldsymbol{y}$ are the data, $\boldsymbol{z}$ are the model variables, and $Q$ is the variational family.

## 2 Derivation of the Posterior distribution

In order to implement any Variational Inference method, it is necessary to explicitly derive the posterior distribution that we aim to approximate.
With this goal in mind, the first step of our work is deriving the joint law of BRAND, together with the full conditionals. For ease of computations we rewrite the model, highlighting the terms $\{\omega_i\}_{i\in\mathbf{N}\setminus\{0\}}$ of the *Stick Breaking* process in terms of the $Beta(1,\gamma)$ that generate them, here called $V_k$.
The joint law is of the form:

$$
\begin{aligned}
p(\mathbf{z},\mathbf{y}) \propto & \prod_{m=1}^{M}\prod_{k=1}^{J}(N(\boldsymbol{y}_m|\boldsymbol{\Theta}_k^{obs}))^{\mathbb{1}_{\xi_m=k}} \cdot \prod_{k=J+1}^{\infty}(N(\boldsymbol{y}_m|\boldsymbol{\Theta}_k^{nov}))^{\mathbb{1}_{\xi_m=k}} \cdot \\
& \cdot \prod_{k=1}^{J} NIW(\boldsymbol{\Theta}_k^{obs}|\boldsymbol{\mu}_k^{\star obs},\nu_k^{\star obs},\lambda_k^{\star obs},\boldsymbol{\Psi}_k^{\star obs}) \cdot \prod_{k=J+1}^{\infty} NIW(\boldsymbol{\Theta}_k^{nov}|\boldsymbol{\mu}_0^{nov},\nu_0^{nov},\lambda_0^{nov},\boldsymbol{\Psi}_0^{nov}) \cdot \\
& \cdot \prod_{m=1}^{M}\prod_{k=1}^{J}(\pi_k)^{\mathbb{1}_{\xi_m=k}} \cdot \prod_{k=J+1}^{\infty}\left(\pi_0\cdot V_{k-J}\cdot\prod_{h=1}^{k-J-1}(1-V_h)\right)^{\mathbb{1}_{\xi_m=k}} \cdot \\
& \cdot \prod_{k=1}^{J}(\pi_k)^{\alpha_k-1}\cdot\prod_{k=1}^{\infty}(1-V_k)^{\gamma-1},
\end{aligned}
$$

where $N(y|\Theta)$ denotes the likelihood of $\Theta$ under a *Gaussian* distribution with realization $y$, and $NIW(\Theta|\dots)$ denotes the likelihood of the parameters of a *Normal-inverse-Wishart* distribution with realization $\Theta=(\boldsymbol{\mu},\Sigma)$.
Consequently, the general terms of the full conditionals are:

$$
\xi_m|- \sim Categorical\begin{cases}\pi_k\cdot N(y_m|\boldsymbol{\Theta}_k^{obs})\cdot\mathbb{1}_{\xi_m=k} & for\ k=1,\ \dots,J,\\ \pi_0\cdot V_k\left(\prod_{l=1}^{k-J-1}(1-V_l)\right)\cdot N(y_m|\boldsymbol{\Theta}_k^{nov})\cdot\mathbb{1}_{\xi_m=k} & for\ k=J+1,\ \dots,\infty,\end{cases}
$$

$$
\boldsymbol{\pi}|- \sim Dirichlet(\{\alpha_k+n_k\}_{k=0}^{J}),
$$

$$
V_k|- \sim Beta\left(n_{J+k}+1,\gamma+\sum_{l=k+1}^{\infty}n_{J+l}\right),
$$

$$
\boldsymbol{\Theta}_k^{obs/nov}|- \sim NIW(\boldsymbol{\mu}_n,\lambda_n,\nu_n,\boldsymbol{\Psi}_n),
$$

where

$$
\begin{aligned}
n_k &= \begin{cases}\#[\xi_m=k] & \text{if } k\geq 1\ ,\\ \#[\xi_m>J] & \text{if } k=0\ ,\end{cases}\\
\boldsymbol{\mu}_n &= \frac{\lambda\cdot\mu_0+\sum_{m=1}^{M}\boldsymbol{y}_m\cdot\mathbb{1}_{\xi_m=k}}{\lambda+n_k},\\
\lambda_n &= \lambda+n_k,\\
\nu_n &= \nu+n_k,\\
\boldsymbol{\Psi}_n &= \boldsymbol{\Psi}_0+\boldsymbol{S}+\frac{\lambda\cdot n_k}{\lambda+n_k}\left(\frac{\sum_{m=1}^{M}\boldsymbol{y}_m\cdot\mathbb{1}_{\xi_m=k}}{n_k}-\boldsymbol{\mu}_0\right)\left(\frac{\sum_{m=1}^{M}\boldsymbol{y}_m\cdot\mathbb{1}_{\xi_m=k}}{n_k}-\boldsymbol{\mu}_0\right)^T,\\
\boldsymbol{S} &= \sum_{m=1}^{M}(\boldsymbol{y}_m-\overline{\boldsymbol{y}}^k)\cdot(\boldsymbol{y}_m-\overline{\boldsymbol{y}}^k)^T\cdot\mathbb{1}_{\xi_m=k},\\
\overline{\boldsymbol{y}}_k &= \frac{\sum_{m=1}^{M}\boldsymbol{y}_m\cdot\mathbb{1}_{\xi_m=k}}{\sum_{m=1}^{M}\mathbb{1}_{\xi_m=k}}.
\end{aligned}
$$

Here, ($\boldsymbol{\mu}_0$, $\lambda_0$, $\nu_0$, $\boldsymbol{\Psi}_0$) indicate the hyperparameters of the relative *NIW* distribution, according to which atom $\boldsymbol{\Theta}_k^{\cdot\cdot}$ is being considered.

# 3 Variational Inference for BRAND

The goal of this section is to adapt variational inference algorithm on the specific case of BRAND. An important source for how to approach the required techniques has been the work conducted by Blei and Jordan in Blei and Jordan [2006].

## 3.1 The choice of variational families for BRAND

The choice of a variational family is undoubtedly one of the most crucial point, since the whole performance of the method depends on it.
In Blei and Jordan [2006] it is suggested to choose the variational family as the same distribution defining the full conditionals, granted they themselves belong to the exponential family. For this reason the chosen variational family is as follows:

$$q(\boldsymbol{\xi}, \boldsymbol{\pi}, \boldsymbol{v}, \boldsymbol{\theta}^{obs}, \boldsymbol{\theta}^{nov}) = q_{\boldsymbol{\eta}}(\boldsymbol{\pi}) \cdot \prod_{m=1}^{M} q_{\boldsymbol{\varphi}^{(m)}}(\xi^{(m)}) \cdot \prod_{k=1}^{T-1} q_{a_k,b_k}(v_k) \cdot$$
$$\cdot \prod_{k=1}^{J} q_{\boldsymbol{\mu}_k^{obs},\lambda_k^{obs},\nu_k^{obs},\boldsymbol{\Psi}_k^{obs}}(\boldsymbol{\theta}_k^{obs}) \cdot \prod_{k=J+1}^{J+T} q_{\boldsymbol{\mu}_k^{nov},\lambda_k^{nov},\nu_k^{nov},\boldsymbol{\Psi}_k^{nov}}(\boldsymbol{\theta}_k^{nov}),$$

where $q_{\boldsymbol{\eta}}(\boldsymbol{\pi})$ is the density function of a *Dirichlet* distribution, $q_{\boldsymbol{\varphi}^{(m)}}(\xi^{(m)})$ of a *Categorical* distribution, $q_{a_k,b_k}(v_k)$ of *Beta* distributions, $q_{\boldsymbol{\mu}_k^{obs},\lambda_k^{obs},\nu_k^{obs},\boldsymbol{\Psi}_k^{obs}}(\boldsymbol{\theta}_k^{obs})$ of *Normal-inverse-Wishart* models, as well as $q_{\boldsymbol{\mu}_k^{nov},\lambda_k^{nov},\nu_k^{nov},\boldsymbol{\Psi}_k^{nov}}(\boldsymbol{\theta}_k^{nov})$.
Moreover, in order to manage the infinite terms derived from the *Dirichlet Process*, we decide to apply a truncation at the first T elements, where T is itself a variational hyperparameter, as in Blei and Jordan [2006].
This truncation can be expressed as $V_T = 1$, hence more rigorously

$$q_T(v_T) = \delta_1(v_T),$$

where $\delta_1(\cdot)$ represents Dirac's delta centered at 1.
From an interpretative standpoint, we approximate the posterior distribution of BRAND in the following parametric form:

$$\begin{aligned}
&\xi_m \overset{ind}{\sim} \sum_{k=1}^{\infty} \varphi_k^{(m)} \cdot \delta_k(\cdot) && m = 1, \ \dots, M, \\
&\boldsymbol{\pi} \sim Dirichlet(\eta_0, \eta_1, ..., \eta_J), && \\
&V_k \overset{ind}{\sim} Beta(a_k, b_k) && k = 1, \ \dots, T-1, \\
&V_T \sim \delta_1(\cdot), && \\
&\boldsymbol{\Theta}_k^{obs} \overset{ind}{\sim} NIW(\boldsymbol{\mu}_k^{obs}, \lambda_k^{obs}, \nu_k^{obs}, \boldsymbol{\Psi}_k^{obs}) && k = 1, \ \dots, J, \\
&\boldsymbol{\Theta}_k^{nov} \overset{ind}{\sim} NIW(\boldsymbol{\mu}_k^{nov}, \lambda_k^{nov}, \nu_k^{nov}, \boldsymbol{\Psi}_k^{nov}) && k = J+1, \ \dots, J+T.
\end{aligned}$$

## 3.2 CAVI parameters update

Now that we have the parametric expression for the variational family, we derive the explicit formulas for the parameter updates in CAVI algorithm.
First of all, thanks to the factorization of the joint law, the update is independent from the terms after the truncation, hence:

$$exp(\mathbb{E}[p]) \propto exp(\mathbb{E}[\log(p_{pre-T})]).$$

Therefore, it is possible to ignore the components of the joint law that appear after the truncation when carrying out the computations.
Moreover, even though the update rule for CAVI involves the whole $\log(p(z_j|\boldsymbol{z}_{-j}, \boldsymbol{y}))$, in practice one can just use the expressions of the full conditionals, since:

$$
\begin{aligned}
exp(\mathbb{E}_q[\log p(z_j|\boldsymbol{z}_{-j},\boldsymbol{y})]) &= exp(\mathbb{E}_q[\log(f(z_j|\boldsymbol{z}_{-j},\boldsymbol{y})\cdot const(\boldsymbol{z}_{-j},\boldsymbol{y}))]) = \\
&= exp(\mathbb{E}_q[\log f(z_j|\boldsymbol{z}_{-j},\boldsymbol{y})] \ + \mathbb{E}_q[\log const(\boldsymbol{z}_{-j},\boldsymbol{y})]) = \\
&= exp(\mathbb{E}_q[\log f(z_j|\boldsymbol{z}_{-j},\boldsymbol{y})])\cdot exp(\mathbb{E}_q[\log const(\boldsymbol{z}_{-j},\boldsymbol{y})]) = \\
&= exp(\mathbb{E}_q[\log f(z_j|\boldsymbol{z}_{-j},\boldsymbol{y})])\cdot const^{\star}(\boldsymbol{z}_{-j},\boldsymbol{y}) \propto \\
&\propto exp(\mathbb{E}_q[\log f(z_j|\boldsymbol{z}_{-j},\boldsymbol{y})]).
\end{aligned}
$$

This step, altought mathematically trivial, is fundamental for easing up the computations of the explicit update rules of CAVI, which are now presented.
Let us start with the law for $\{\eta_j\}_{j=0}^{J}$:

$$
\eta_j = \begin{cases} \alpha_j + \sum_{m=1}^{M}\sum_{l=J+1}^{J+T}\varphi_l^{(m)} & for\ j=0, \\ \alpha_j + \sum_{m=1}^{M}\varphi_j^{(m)} & for\ j=1,\ \dots,J. \end{cases}
$$

One can notice that we are updating the parameter of the *Dirichlet* distribution linked to each known component with the sum of the probability of all data belonging to that specific component. Likewise, for the parameter linked with the novelty, $\eta_0$, we use the sum of the probabilities of all data belonging to the novelty.
The update for $\{(a_k,b_k)\}_{k=1}^{T-1}$ is given by:

$$
a_k = 1 + \sum_{m=1}^{M}\varphi_{J+k}^{(m)},
$$

$$
b_k = \gamma + \sum_{l=k+1}^{T}\sum_{m=1}^{M}\varphi_{J+l}^{(m)}.
$$

This update can be translated in adjusting the first parameter of the *Beta* distribution tied to a specific novelty cluster with the probability of each data to belong to said cluster, while we are updating the second parameter with the probability of belonging to the next novelty components, hence to the probability of not having yet been assigned to a cluster.
The update for parameters $\{(\boldsymbol{\mu}_k^{obs},\lambda_k^{obs},\nu_k^{obs},\boldsymbol{\Psi}_k^{obs})\}_{k=1}^{J}$ and $\{(\boldsymbol{\mu}_k^{nov},\lambda_k^{nov},\nu_k^{nov},\boldsymbol{\Psi}_k^{nov})\}_{k=J+1}^{J+T}$ is given, for both known and novelty components, according to:

$$
\boldsymbol{\mu}_k = \frac{\lambda_0\cdot\boldsymbol{\mu}_0 + \sum_{m=1}^{M}\boldsymbol{y}_m\cdot\varphi_k^{(m)}}{\lambda_0 + \sum_{m=1}^{M}\varphi_k^{(m)}},
$$

$$
\lambda_k = \lambda_0 + \sum_{m=1}^{M}\varphi_k^{(m)},
$$

$$
\nu_k = \nu_0 + \sum_{m=1}^{M}\varphi_k^{(m)},
$$

$$
\boldsymbol{\Psi}_k = \boldsymbol{\Psi}_0 + \sum_{m=1}^{M}(\boldsymbol{y}_m-\overline{\boldsymbol{y}}_k)\cdot(\boldsymbol{y}_m-\overline{\boldsymbol{y}}_k)^T\cdot\varphi_k^{(m)} + \frac{\lambda_0\cdot\sum_{m=1}^{M}\varphi_k^{(m)}}{\lambda_0+\sum_{m=1}^{M}\varphi_k^{(m)}}\cdot(\overline{\boldsymbol{y}}_k-\boldsymbol{\mu}_0)^T\cdot(\overline{\boldsymbol{y}}_k-\boldsymbol{\mu}_0),
$$

$$
\overline{\boldsymbol{y}}_k = \frac{\sum_{m=1}^{M}\boldsymbol{y}_m\cdot\varphi_k^{(m)}}{\sum_{m=1}^{M}\varphi_k^{(m)}}.
$$

The interpretation of this updating rule is more challenging, but one could draw a strong parallelism with the posterior distribution of the conjugate model *Normal Normal-inverse-Wishart*, indeed for instance $\overline{\boldsymbol{y}}_k$ is just a weighted mean of all data, where weights are derived from the probability of belonging to said cluster.
The last update rule is the one of parameters $\{\varphi_k^{(m)}\}_{k=1\ m=1}^{J+T\ M}$:

$$\varphi_k^{(m)} \propto \begin{cases} exp\{\mathbb{E}[\log \pi_k] + \mathbb{E}[\log N(\boldsymbol{y}_m|\boldsymbol{\Theta}_k^{obs})]\} & for\ k = 1,\ \dots, J, \\ exp\{\mathbb{E}[\log \pi_0] + \mathbb{E}[\log V_{k-J}] + \sum_{l=1}^{k-J-1} \mathbb{E}[\log 1 - V_l] + \mathbb{E}[\log N(\boldsymbol{y}_m|\boldsymbol{\Theta}_k^{nov})]\} & for\ k = J+1,\ \dots, J+T. \end{cases}$$

This means that the probability of datum $\boldsymbol{y}_m$ to belong to cluster $k$ depends on the likelihood of $\boldsymbol{y}_m$ under that same cluster and on the overall relevance of $k^{th}$ cluster. Such relevance is determined as the expected value of the relative component of the *Dirichlet* distributed $\boldsymbol{\pi}$ and, for novelties, the *Stick Breaking* parameter, here unrolled in its *Beta* distributed components.
Here we report the explicit expression of the terms of this last updating rule:

$$\mathbb{E}[\log \pi_k] = \boldsymbol{\psi}(\eta_k) - \boldsymbol{\psi}(\overline{\eta}),$$

$$\mathbb{E}[\log V_{k-J}] = \boldsymbol{\psi}(a_{k-J}) - \boldsymbol{\psi}(a_{k-J} + b_{k-J}),$$
$$\mathbb{E}[\log (1 - V_l)] = \boldsymbol{\psi}(b_l) - \boldsymbol{\psi}(a_l + b_l),$$

$$\mathbb{E}[\log N(\boldsymbol{y}_m|\boldsymbol{\Theta}_k^{\cdot\cdot})] = \frac{1}{2} \cdot \left( -p \cdot \ln 2\pi + \sum_{l=1}^{p} \boldsymbol{\psi}\left(\frac{\nu_k - l + 1}{2}\right) + p \cdot \ln 2 + \ln |\boldsymbol{\Psi}_k^{-1}| \right) +$$
$$- \frac{1}{2} \cdot \left( \frac{p}{\lambda_k} + \nu_k \cdot (\boldsymbol{y}_m - \boldsymbol{\mu}_k)^T \boldsymbol{\Psi}_k^{-1} (\boldsymbol{y}_m - \boldsymbol{\mu}_k) \right).$$

Indeed, exploiting the results on the expected value of the log determinant of a *Wishart* distributed random variable, we have that:

$$\mathbb{E}[\log |\Sigma_\theta^{-1}|] = -p \cdot \ln 2 - \ln |\boldsymbol{\Psi}_k^{-1}| - \sum_{l=1}^{p} \boldsymbol{\psi}\left(\frac{\nu_k - l + 1}{2}\right).$$

### 3.3 ELBO computation

Finally, there is the need of an explicit way to evaluate the ELBO, as it is the metric used to check the convergence of the optimization algorithm.
We recall that:

$$ELBO(q) = \mathbb{E}[\log p(\boldsymbol{z}, \boldsymbol{y})] - \mathbb{E}[\log q(\boldsymbol{z})].$$

Decomposing the previous formula element-wise:

$$\mathbb{E}[\log p] = \sum_{m=1}^{M} \left( \sum_{k=1}^{J} (f_1^{(m,k)}) + \sum_{k=J+1}^{J+T} (f_2^{(k)}) \right) + \sum_{k=1}^{J} (f_3^{(k)}) + \sum_{k=J+1}^{J+T} (f_4^{(k)}) +$$
$$+ \sum_{m=1}^{M} \left( \sum_{k=1}^{J} (f_5^{(m,k)}) + \sum_{k=J+1}^{J+T} (f_6^{(m,k)}) \right) + \sum_{k=0}^{J} (f_7^{(k)}) + \sum_{l=1}^{T} (f_8^{(l)}) + const,$$

where:

$$f_1^{(m,k)} = \varphi_k^{(m)} \cdot \mathbb{E}[\log N(\boldsymbol{y}_m|\boldsymbol{\Theta}_k^{obs})],$$
$$f_2^{(m,k)} = \varphi_k^{(m)} \cdot \mathbb{E}[\log N(\boldsymbol{y}_m|\boldsymbol{\Theta}_k^{nov})],$$
$$f_3^{(m,k)} = \mathbb{E}[\log NIW(\boldsymbol{\Theta}_k^{obs}|\boldsymbol{\mu}_k^{\star obs}, \nu_k^{\star obs}, \lambda_k^{\star obs}, \boldsymbol{\Psi}_k^{\star obs})],$$
$$f_4^{(m,k)} = \mathbb{E}[\log NIW(\boldsymbol{\Theta}_k^{nov}|(\boldsymbol{\mu}_0^{nov}, \nu_0^{nov}, \lambda_0^{nov}, \boldsymbol{\Psi}_0^{nov})],$$
$$f_5^{(m,k)} = \varphi_k^{(m)} \cdot (\boldsymbol{\psi}(\eta_k) - \boldsymbol{\psi}(\sum_{j=0}^{J} \eta_j)),$$
$$f_6^{(m,k)} = \varphi_k^{(m)} \cdot \left( \boldsymbol{\psi}(\eta_0) - \boldsymbol{\psi}(\sum_{j=0}^{J} \eta_j) + \boldsymbol{\psi}(a_{k-J} + b_{k-J}) + \sum_{h=1}^{k-J-1} (\boldsymbol{\psi}(b_h) - \boldsymbol{\psi}(a_h + b_h)) \right),$$

$$f_7^{(k)} = (\alpha_k - 1) \cdot (\boldsymbol{\psi}(\eta_k) - \boldsymbol{\psi}(\sum_{j=0}^{J} \eta_j)),$$

$$f_8^{(m,k)} = (\gamma - 1) \cdot (\boldsymbol{\psi}(b_l) - \boldsymbol{\psi}(a_l + b_l)),$$

and

$$\mathbb{E}[\log N(\boldsymbol{y}_m|\boldsymbol{\Theta}_k^{\cdot\cdot})] = \frac{1}{2} \cdot \left( \log|\boldsymbol{\Psi}_k^{-1}| + \sum_{i=1}^{p} \psi(\frac{\nu_k - i + 1}{2}) - \frac{p}{\lambda_k} - \nu_k \cdot (\boldsymbol{y}_m - \boldsymbol{\mu}_k)^T \boldsymbol{\Psi}_k^{-1} (\boldsymbol{y}_m - \boldsymbol{\mu}_k) \right) + const,$$

$$\begin{aligned} \mathbb{E}[\log(NIW(\boldsymbol{\Theta}_k^{\cdot\cdot}|\mu_{\mathbf{0}}, \lambda_{\mathbf{0}}, \nu_{\mathbf{0}}, \boldsymbol{\Psi}_{\mathbf{0}}))] \; =& \frac{1}{2} \cdot \big( p \cdot \ln \frac{\lambda_0}{2\pi} + \sum_{i=1}^{p} \psi(\frac{\nu_k - i + 1}{2}) + \ln|\boldsymbol{\Psi}_k^{-1}| + \\ & - p \cdot \frac{\lambda_0}{\lambda_k} - \lambda_0 \cdot \nu_k \cdot (\boldsymbol{\mu}_k - \boldsymbol{\mu}_0)^T \boldsymbol{\Psi}_k^{-1} (\boldsymbol{\mu}_k - \boldsymbol{\mu}_0) \big) + \\ & + \ln B(\boldsymbol{\Psi}_0^{-1}, \nu_0) - \frac{1}{2} \cdot \nu_k \cdot Tr(\boldsymbol{\Psi}_0 \cdot \boldsymbol{\Psi}_k^{-1}) + const, \end{aligned}$$

with

$$B(\boldsymbol{\Psi}_0^{-1}, \nu_0) = |\boldsymbol{\Psi}_0^{-1}|^{\frac{-\nu_0}{2}} \cdot \left( 2^{\frac{p \cdot \nu_0}{2}} \cdot \pi^{\frac{p \cdot (p-1)}{4}} \cdot \prod_{i=1}^{p} \Gamma\left( \frac{\nu_0 - i + 1}{2} \right) \right)^{-1}.$$

For the second term:

$$\mathbb{E}[\log q] = \sum_{m=1}^{M} (h_1^{(m)} + h_2^{(m)} + \sum_{k=1}^{T} h_3^{(m,k)} + \sum_{k=1}^{J} h_4^{(m,k)} + \sum_{k=1}^{T} h_5^{(m,k)}),$$

with

$$h_1^{(m)} = \sum_{k=1}^{J+T} \varphi_k^{(m)} \cdot \ln \varphi_k^{(m)} - \ln \sum_{k=0}^{J+T} \varphi_k^{(m)},$$

$$h_2^{(m)} = \sum_{j=0}^{J} (\eta_j - 1) \cdot \left( \boldsymbol{\psi}(\eta_j) - \boldsymbol{\psi}(\sum_{j=0}^{J} \eta_j) \right) + \ln \Gamma \left( \sum_{j=0}^{J} \eta_j \right) - \sum_{j=0}^{J} \ln \Gamma(\eta_j),$$

$$h_3^{(m,k)} = (a_k - 1) \cdot (\boldsymbol{\psi}(a_k) - \boldsymbol{\psi}(a_k + b_k)) + (b_k - 1) \cdot (\boldsymbol{\psi}(b_k) - \boldsymbol{\psi}(a_k + b_k)) - \ln \left( \frac{\Gamma(a_k) \cdot \Gamma(b_k)}{\Gamma(a_k + b_k)} \right),$$

$$h_4^{(m,k)} = \mathbb{E}[\log NIW(\boldsymbol{\Theta}_k^{obs}|\boldsymbol{\mu}_k^{obs}, \lambda_k^{obs}, \nu_k^{obs}, \boldsymbol{\Psi}_k^{obs})],$$

$$h_5^{(m,k)} = \mathbb{E}[\log NIW(\boldsymbol{\Theta}_k^{nov}|\boldsymbol{\mu}_k^{nov}, \lambda_k^{nov}, \nu_k^{nov}, \boldsymbol{\Psi}_k^{nov})],$$

where

$$\begin{aligned} \mathbb{E}[\log(NIW(\boldsymbol{\Theta}_k^{\cdot\cdot}|\boldsymbol{\mu}_k, \lambda_k, \nu_k, \boldsymbol{\Psi}_k))] \; =& \frac{\nu_k}{2} \cdot \ln|\boldsymbol{\Psi}_k| - \frac{\nu_k \cdot p}{2} \cdot \ln 2 - \Gamma_p \left( \frac{\nu_k}{2} \right) + \frac{p}{2} \cdot \log \lambda_k + \\ & + \frac{\nu_k + p + 2}{2} \cdot \left( \log|\boldsymbol{\Psi}_k^{-1}| + \sum_{i=1}^{p} \boldsymbol{\psi} \left( \frac{\nu_k - i + 1}{2} \right) \right) - \frac{\nu_k}{2} \cdot p. \end{aligned}$$

# 4 Implementation in *Python*

To test this procedure and its effectiveness we implement the algorithm and measure its performance on simulated data.
The selected environment for the embedding of the procedure is *Python*, this choice is the result of a trade off between ease of implementation and computational efficiency. Remembering that the whole goal is to develop an efficient alternative to MCMC, performances are boosted through the exploitation of the *JAX* library.
The two-stage procedure of the method are again reflected in the code structure. We can identify two core elements:

- First stage code: receives as input the training set, including labels of the observed classes. Computes MRCD estimators for location and dispersion parameters.
- Second stage code: receives as input the test set, the hyperparameter specification for the Semi-parametric Bayesian Mixture, the estimators computed at the first stage, the required tolerance to determine ELBO convergence (together with a maximum number of iterations) and, optionally, an user-defined initialization for CAVI algorithm.
  CAVI is applied in order to compute the approximation of the posterior distribution. Clusters are defined by considering the *argmax* over the cluster assignment variables distribution, hence considering the maximum a posteriori probability of belonging to each population.

This procedure is summarized in Figure 1; the produced code can be found at https://github.com/pietrocipolla/VariationalBRAND.

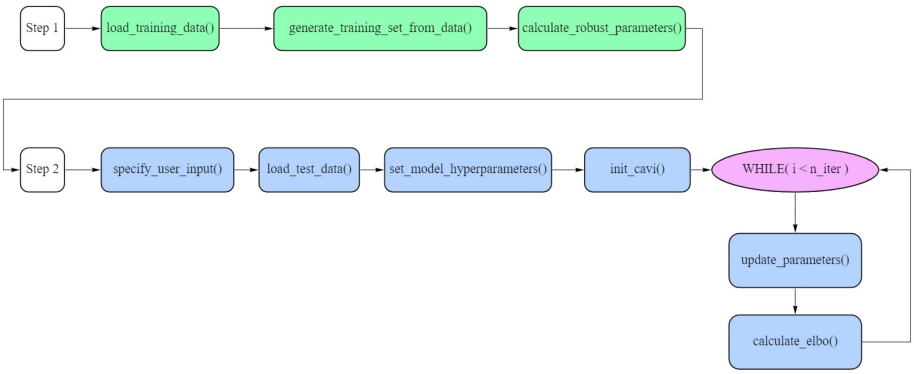


Figure 1: Code pipeline

The first simulated scenario we use for testing the code is composed of five bidimensional classes, of which two are novelty. The class composition in the two datasets is reported in Table 1:

| Population | Cardinality - Training Set | Cardinality - Test Set |
|---|---|---|
| $Observed_1$ | 200 | 200 |
| $Observed_2$ | 250 | 250 |
| $Observed_3$ | 200 | 200 |
| $Novelty_1$ | 0 | 90 |
| $Novelty_2$ | 0 | 50 |

Table 1: Class composition in the adopted simulated scenario

These populations are distributed according to a *Multivariate Gaussian* with the identity as a covariance matrix.
As mentioned before, the partition on the test set is defined by:

$$\rho^*(\boldsymbol{y}_m) = \underset{k \in \{1,\dots J+T\}}{\arg\max}\ \varphi_k^{(m)}$$

hence by considering each datum as belonging to the class with the highest estimated a posteriori probability.
In Figure 2 it is reported the partition induced, obtained with a truncation of the *Dirichlet Process* equal to $T = 20$, a tolerance on ELBO $\varepsilon = 10^{-3}$ and using the prior distributions for initializing CAVI.

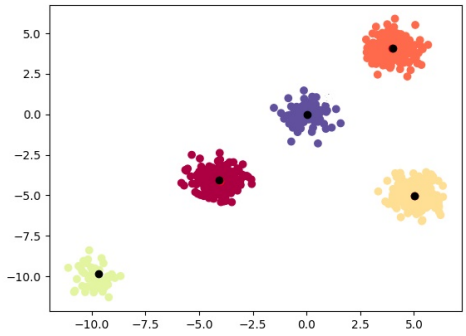


Figure 2: Partition induced on test data

The points highlighted in black are the centers of each found group.
It is clear how this model is successfully able to detect novelties, while still correctly classifying the observations coming from already known classes.
It should however be mentioned that, since the Variational approach consists in an optimization procedure, it suffers of the same difficulties characteristics of this discipline.
In particular we remind that there is the possibility to reach a locally optimal solution which maximizes ELBO, resulting in a sub-optimal approximation of the posterior distribution. This problem is heavily connected with initialization sensitivity.
To address this issue, an effective solution is to randomize the initialization of CAVI, especially for what concerns the centers of the novelties $\{\boldsymbol{\mu}_k^{nov}\}$. A possibility would be to run multiple instances in parallel, with different random initialization, and keeping only the one which yields the maximum ELBO.
In this article, for efficiency's sake, we instead try to propose a single, data driven but still not informative initialization: the $T$ centers of the novelty populations are initialized according to the centers of a k-means clustering, with $T$ populations, over the test set.
The trials conducted support the effectiveness of this initialization, which allows for data to be immediately grouped amongst novelties too, limiting the occurrence of novelties being grouped in a single, widespread cluster, a local maximum which frequently arise during simulations.
Moreover, it should be specified that this procedure does not require any knowledge of the actual composition of the test set, indeed in practice the truncation $T$ is chosen as a big overestimation of the actual amount of novelties, hence the $\{\boldsymbol{\mu}_k^{nov}\}$ will be initialized not only on top of observed populations as well, but it is also possible that multiple centers are competing for the same population, it will be then up to CAVI to re-absorb known components into the observed populations and to trim down the excess novelties.

# 5 Performance comparison against MCMC

We originally adopted a variational Bayes approach to tackle problems that are practically intractable, either because of time inefficiency or a too big computational effort, by a standard MCMC based approach.
To prove the gain in efficiency provided by our method, we perform a small simulation study comparing the performance of the variational approach, described above, against an MCMC strategy.
In particular, we compare the performances of our *JAX* implementation with the *Rcpp* routines proposed in Denti and Cappozzo [2021], in which the second stage is implemented via Gibbs Sampler.
It should be stressed that *Rcpp* allows the most computationally expensive parts to be implemented in *C++*, which is a much more efficient environment than *Python*; since we aimed for a more balanced trade off between ease of implementation and computational efficiency, it is likely that with this trial we are underestimating the efficiency of our proposal for a procedure.
The reference scenario for this comparison consists in a bidimensional dataset with seven *Normal* random variables, of which four are interpreted as novelties. Moreover, two of these novelties intersect. Figure 3 showcases the simulated dataset, and Table 2 describes the clusters' frequencies.

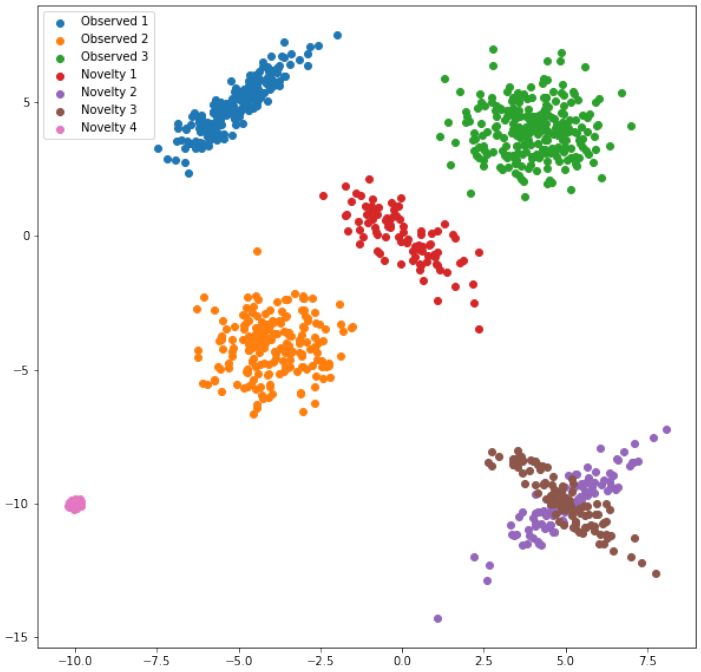


Figure 3: Reference Test Set

| Population | Cardinality Training Set | Cardinality Test Set |
|---|---|---|
| $Observed_1$ | 300 | 200 |
| $Observed_2$ | 300 | 200 |
| $Observed_3$ | 400 | 250 |
| $Novelty_1$ | 0 | 90 |
| $Novelty_2$ | 0 | 100 |
| $Novelty_3$ | 0 | 100 |
| $Novelty_4$ | 0 | 60 |
| Total | 1000 | 1000 |

Table 2: Class composition in reference dataset ($n = 1000$, $p = 2$)

We fit multiple models for different combinations of sample size $n$ and data dimensionality $p$. The dimensionality augmentation is carried out by adding, for each datum and for each extra dimension,

a realization from a $N(0,1)$, independently both between and within each datum.
For example, for a generic population in $\mathbb{R}^2$ defined as:

$$\boldsymbol{y} \overset{iid}{\sim} N(\boldsymbol{\mu}, \Sigma)$$

its $\mathbb{R}^p$, $p > 2$, augmentation is defined as:

$$\boldsymbol{y} \overset{iid}{\sim} N\left(\begin{pmatrix}\boldsymbol{\mu} \\ \mathbf{0}_{\{p-2\}}\end{pmatrix}, \begin{pmatrix}\Sigma & 0 \\ 0 & \mathbb{I}_{\{p-2,p-2\}}\end{pmatrix}\right)$$

Note that this dimensionality augmentation approach does not induce any additional separation of clusters, the actual partition is still uniquely identified by the first two dimensions and all the extra ones are just for performance comparison.
To preserve class balances during the trials, when modifying the total sample size, population proportions are preserved.
The comparison is performed on the grid defined over $n \in \{500, 1000, 2500, 5000, 10000\}$ and $p \in \{2, 3, 5, 7, 10\}$.
To conduct the study, 10 couples of datasets for each element $(n, p)$ of the grid are generated, and for each scenario the comparison is carried out between:

- The MCMC based implementation, with 80'000 samples of which 20'000 are *burn-in* iterations.
- The Variational Inference based implementation, with a tolerance $\varepsilon = 10^{-5}$ and initialization given by k-means.

In both cases, the truncation of the *Dirichlet Process* is set at $T = 20$.
The proposed metric to check the quality of the estimated clustering is the *Adjusted Rand Index* (ARI), computed over all data with respect to their true classification, without taking into account whether an observation comes from an already observed population or not. Therefore, our adopted metric evaluates the overall clustering induced on the test set.
By definition, ARI between two clustering structures $(\rho, \rho^\star)$ is computed as follows:

$$ARI(\rho, \rho^\star) = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}$$

defining the induced partitions as $\rho = (C_1, \dots C_N)$ and $\rho^\star = (C_1^\star, \dots C_{N^\star}^\star)$, $n_{i,j} = \#[C_i \cap C_j^\star]$, $a_i = \#[C_i]$ and $b_j = \#[C_j^\star]$. Notice that it is not required for $N$ to be equal to $N^\star$
For reference, the machine on which this simulations have been carried out has the following characteristics:

- Processor: Intel Core i5-8250U CPU, 1.60GHz x 4.
- Memory: 7.7 GiB.

In Table 3, we report the results for the trials conducted on the MCMC based procedure.

| $(n, p)$ | Mean computation time | Mean ARI |
|---|---|---|
| (500,2) | 3.695 min | 0.932 |
| (500,3) | 4.033 min | 0.922 |
| (500,5) | **out of memory error** | |
| (1000,2) | 6.685 min | 0.930 |
| (1000,3) | 7.530 min | 0.931 |
| (1000,5) | **out of memory error** | |

Table 3: Summary of MCMC performances

The MCMC based approach leads to particularly accurate classifications, but as soon as the dataset increases its size, either in the number of samples or in their dimensionality, it becomes unfeasible due to the rapidly increasing amount of memory required to store all the MCMC samples.
The data structure to be stored at each sample is indeed relatively large and in the failed attempts reaches sizes in the order of $\sim 5\ GiB$ after few thousand samples.
Such a problem renders this approach too complex to be performed on a standard laptop like the one used to carry out these simulations.
In Table 4, we report the results for the trials conducted on the Variational Inference based procedure.

| $(n, p)$ | Mean n° iters performed | Mean computation time | Mean time per iteration | Mean ARI |
|---|---|---|---|---|
| (500,2) | 43.7 | 33.784 s | 0.773 s/iter | 0.880 |
| (500,3) | 50 | 40.612 s | 0.812 s/iter | 0.847 |
| (500,5) | 58.1 | 49.437 s | 0.851 s/iter | 0.823 |
| (500,7) | 24.4 | 23.614 s | 0.968 s/iter | 0.744 |
| (500,10) | 37.8 | 40.728 s | 1.077 s/iter | 0.514 |
| (1000,2) | 54.2 | 63.777 s | 1.177 s/iter | 0.906 |
| (1000,3) | 50 | 69.726 s | 1.394 s/iter | 0.892 |
| (1000,5) | 62.7 | 94.622 s | 1.509 s/iter | 0.836 |
| (1000,7) | 52.2 | 89.609 s | 1.717 s/iter | 0.821 |
| (1000,10) | 24.7 | 70.033 s | 2.835 s/iter | 0.575 |
| (2500,2) | 75.6 | 348.311 s (5.8 min) | 5.084 s/iter | 0.965 |
| (2500,3) | 80.4 | 480.341 s (8.0 min) | 5.974 s/iter | 0.928 |
| (2500,5) | 70.5 | 546.396 s (9.1 min) | 8.006 s/iter | 0.846 |
| (2500,7) | 61.9 | 605.175 s (10.1 min) | 9.777 s/iter | 0.838 |
| (2500,10) | 30.1 | 406.867 s (6.8 min) | 13.517 s/iter | 0.640 |
| (5000,2) | 109.9 | 1932.017 s (32.2 min) | 17.580 s/iter | 0.956 |
| (5000,3) | 116.7 | 2574.669 s (42.9 min) | 22.062 s/iter | 0.966 |
| (5000,5) | 135.7 | 3742.641 s (62.4 min) | 27.580 s/iter | 0.848 |
| (5000,7) | 146.3 | 5063.884 s (84.4 min) | 34.613 s/iter | 0.845 |
| (5000,10) | 55.3 | 2685.851 s (44.8 min) | 48.569 s/iter | 0.724 |
| (10000,2)$\star$ | 74 | 2797.070 s (46.6 min) | 37.798 s/iter | 0.859 |
| (10000,3)$\star$ | 53 | 2539.277 s (42.3 min) | 47.911 s/iter | 0.891 |
| (10000,5)$\star$ | 36 | 2181.104 s (36.4 min) | 60.586 s/iter | 0.852 |
| (10000,7)$\star$ | **out of memory error** | | | |

Table 4: Summary of Variational Inference performances

Due to the extensive time required to run each simulation, the results denoted with $\star$ are computed over a single realization of the dataset.
The Variational approach leads to analogous performances in terms of yielded classification, but with significantly smaller computational and time requirements. Such efficiency allows for the processing of problems which would have been unfeasible for an MCMC based approach.
Due to the variability in the number of CAVI iterations required to reach convergence with respect to a given tolerance, measuring the performances purely on the overall time required would lead to unreliable conclusions if the other factors at play are not adequately taken into account. For this reason, we propose the time required for a single CAVI iteration as a metric of performance for what concerns the efficiency of the variational approach to BRAND, the observed behaviour of this metric is plotted in Figure 4.
It should be stressed that the main advantage of this procedure, for what concerns its applicability, is that it is no longer needed to store a complex data structure for tens of thousands of samples; at each iteration there is the need for just and update of a fixed size data structure.

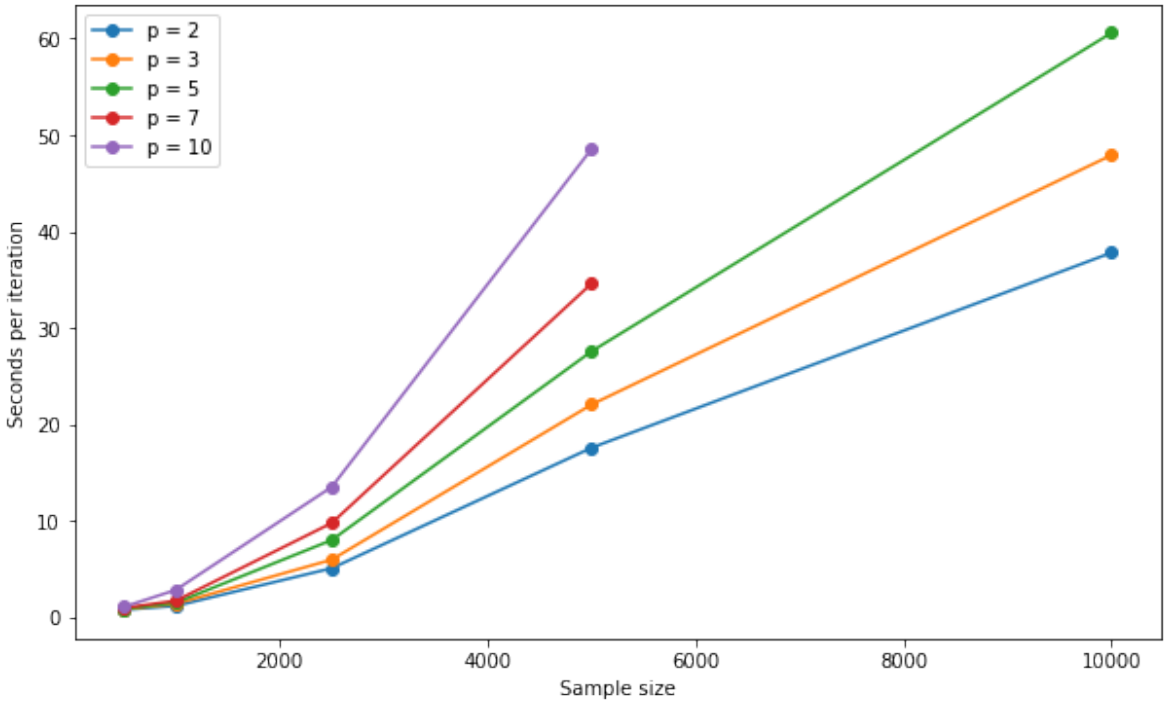


Figure 4: Trends of seconds per CAVI iteration

# 6 Conclusions

Novelty detection is a complicated task, especially given the high complexity of the model used to tackle it. The need of a semiparametric model renders MCMC based inference particularly demanding in terms of memory and computational time, especially with large dataset.

We effectively manage to solve this issue by applying a variational inference approach in place of a sampling strategy. This not only translates in a substantially smaller computation time, but also in much more efficient memory storage requirements. Our approach represents a less demanding procedure which is more reasonably feasible, even on regular laptops.

However, this approach retains some limitations. Although the challenges faced by MCMC strategies arise on much larger scenarios, they still do appear and consist a limit in the applicability of BRAND over really large datasets.

Even within the Variational Inference framework, the problem does not have an easy fix. A potential solution can consist in simplifying the model itself, for instance reducing the degrees of freedom of the covariance structures within clusters. For example, one could model such matrices via the identity multiplied by an *inverse-Gamma* distributed random variable.

Nonetheless, this choice will lead to a less accurate clustering procedure which is likely to fail at distinguishing between intersecting populations, like the X-shaped populations which were instead correctly detected in the simulation studies. However, in a truly high-dimensional framework such simplification could become necessary to even approach the problem.

Overall, the results shown in this report illustrate how computational limits and memory constraints are not an immovable barrier when developing a Bayesian procedure. In fact, we were able to effectively reduce the computational cost by approaching the posterior inference problem from another point of view: optimizing a discrepancy problem involving the posterior rather than sampling from it.

# References

David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Francesco Denti and Andrea Cappozzo. Brand. https://github.com/Fradenti/Brand, 2021.

Francesco Denti, Andrea Cappozzo, and Francesca Greselin. A two-stage bayesian semiparametric model for novelty detection with robust prior information. *Statistics and Computing*, 31(4):1–19, 2021.