Int. J. Appl. Math. Comput. Sci., 2021, Vol. 31, No. 3, 487–500
DOI: 10.34768/amcs-2021-0033

FITTING A GAUSSIAN MIXTURE MODEL THROUGH THE GINI INDEX

ADRIANA LAURA LÓPEZ-LOBATO [a,*], MARTHA LORENA AVENDAÑO-GARRIDO [a]

[a] Faculty of Mathematics
University of Veracruz
Circuito Gonzalo Aguirre Beltrán S/N, Zona Universitaria, Xalapa, Veracruz, Mexico
e-mail: adrilau17@gmail.com

A linear combination of Gaussian components is known as a Gaussian mixture model. It is widely used in data mining and pattern recognition. In this paper, we propose a method to estimate the parameters of the density function given by a Gaussian mixture model. Our proposal is based on the Gini index, a methodology to measure the inequality degree between two probability distributions, and consists in minimizing the Gini index between an empirical distribution for the data and a Gaussian mixture model. We will show several simulated examples and real data examples, observing some of the properties of the proposed method.

Keywords: Gini index problem, Gaussian mixture model, clustering.

1. Introduction

Consider the problem of finding clusters for the data set $P = \{p_1, p_2, \ldots, p_M\}$ with $p_m \in \mathbb{R}^N = X$. When we want to analyze the data set by modelling their behaviour, we usually use some of the known density functions, for example the multivariate normal density of dimension N given by

$$f(x|\mu, \Sigma) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}, \quad (1)$$

where μ is the mean and Σ is the covariance matrix. However, sometimes a unimodal distribution cannot represent the information given by the data when there are clusters, so multimodal distributions are used.

In this paper we consider a multimodal distribution that is a linear combination of Gaussian components to model the data, known as a Gaussian mixture model, (Bishop, 2006; Reynolds, 2009). The Gaussian mixture model is widely used for segmentation of images, speech recognition, language identification and statistical representation (Greenspan *et al.*, 2006; Povey *et al.*, 2011; Torres-Carrasquillo *et al.*, 2002; Singh *et al.*, 2009).

In the Gaussian mixture model we consider a density function that is a linear combination of K Gaussian densities of the form

$$\sum_{k=1}^{K} \phi_k f(x|\mu_k, \Sigma_k).$$

The components of this mixture are Gaussian densities $f(x \mid \mu_k, \Sigma_k)$ with mean μ_k and covariance matrix Σ_k, and the ϕ_k parameters are mixing coefficients that must comply with

$$0 \leq \phi_k \leq 1, \quad k = 1, \ldots, K, \quad (2)$$

and

$$\sum_{k=1}^{K} \phi_k = 1. \quad (3)$$

The sought parameters in the Gaussian mixture distribution are $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_K)$, $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_K)$ and $\boldsymbol{\Sigma} = (\Sigma_1, \Sigma_2, \ldots, \Sigma_K)$. One way to set the values of these parameters is to use the maximum likelihood method. The logarithm of the likelihood function for this problem, when we have M elements, that is, $\{p_m\}_{m=1}^{M}$, is given by

$$l(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}|P) = \sum_{m=1}^{M} \log \left(\sum_{k=1}^{K} \phi_k f(p_m \mid \mu_k, \Sigma_k) \right).$$

There is no analytical solution to this problem, so iterative numerical optimization techniques are used for

*Corresponding author

this purpose. In several texts (e.g., Dempster *et al.*, 1977; Meng and Rubin, 1994; Vaida, 2005; Xu and Jordan, 1996), the authors employ a powerful framework called expectation-maximization for Gaussian mixtures. They want to maximize the likelihood function with respect to the parameters $\boldsymbol{\phi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ by following the EM algorithm.

Once the desired $\boldsymbol{\phi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ parameters are obtained, we can perform data classification, using the total probability law and the Bayes theorem. We can obtain the probability that, given a data point x, it belongs to the parametric distribution g_k, that is, $\Pr(g_k|x)$, for $k = 1, \ldots, K$, in the following way:

$$\Pr(g_{k'}|x) = \frac{\Pr(g_{k'})\Pr(x|g_{k'})}{\sum_{k=1}^{K} \Pr(g_k)\Pr(x|g_k)},$$

for $k' = 1, 2, \ldots, K$, where $\Pr(x|g_k)$ is the probability that x comes from the parametric distribution k and $\Pr(g_k) = \phi_k$ is the probability of the parametric distribution k. Once we obtain the probabilities $\Pr(g_k|x)$, with $k = 1, \ldots, K$, we make a comparison of these values and we determine that the point x belongs to the parametric distribution with the greatest value $\Pr(g_k|x)$. In this work we propose an alternative to the EM algorithm, called the GI algorithm, based on minimizing the Gini index between the empirical distribution of the observed data and a mixture of Gaussians.

In Section 2, we will give a brief introduction to the Gini index. In Section 3, we will show the proposed procedure to estimate the parameters of a Gaussian mixture model through the Gini index problem in an efficient way, similarly to but in greater detail than in our previous work (López-Lobato and Avendaño-Garrido, 2020). In Section 4 we will perform exhaustive experiments with simulated data and a few real data sets, with the purpose of comparing the numerical results obtained by the EM algorithm, the K-means method and the algorithm proposed in this work. We will end by giving conclusions and mentioning future work in Section 5.

2. Gini index

The Gini index is a distance between two probability distributions, so it can be used to measure the inequality level between them. This measure is applied as an indicator of social and economic inequality when the income distribution of a country is analyzed and in other several fields like engineering, transport and ecology (see Giorgi and Gigliarano, 2017; Rachev *et al.*, 2013; Ultsch and Lötsch, 2017).

For the Gini index (GI) problem, we consider a space X, two probability distributions ν_1 and ν_2 on X, and a distance function in X, $d : X \times X \to \mathbb{R}$.

The GI problem is as follows:

Minimize

$$\int_{X\times X} d(x, y)\, \mathrm{d}\pi$$

subject to

$$\begin{aligned} \pi(A \times X) &= \nu_1(A), \\ \pi(X \times A) &= \nu_2(A), \\ \pi &\in \mathcal{M}^+(X \times X), \end{aligned}$$

for all A in the Borel σ-algebra of X, where $\mathcal{M}^+(X \times X)$ is the convex cone of probability measures, i.e., π is in the set of joint probability distributions on $X \times X$, whose marginals in the first and second components are the probability distributions ν_1 and ν_2, respectively, denoted by $\Pi(\nu_1, \nu_2)$.

Rubner *et al.* (2000) and Villani (2003) showed that this problem always has a solution, which is a distance between the probability distributions ν_1 and ν_2 and can be very expensive to find. In addition, the solution $\pi^* \in \Pi(\nu_1, \nu_2)$ is a probability measure and the optimal value of this problem defines the Gini index between the probability measures ν_1 and ν_2, denoted by $GI(\nu_1, \nu_2)$, that is,

$$\mathrm{GI}(\nu_1, \nu_2) = \int_{X\times X} d(x, y)\, \mathrm{d}\pi^*.$$

In this paper, we propose to find the Gaussian mixture distribution that minimizes the Gini index to an empirical distribution. We make this proposal based on the work of Bassetti *et al.* (2006), where the theory of the minimum dissimilarity estimators and the estimators of the minimum distance of Kantorovich are discussed. In this type of problems, the distribution ν_1 is known, commonly associated with an empirical distribution, and the distribution ν_2 must be estimated in such a way that the distance between ν_1 and ν_2, given by the Gini index, is minimum. We propose this in order to efficiently estimate the parameters of a Gaussian mixture model through the Gini index problem.

Now, we will explain the process that we follow to establish the Gini Index problem based on a given data set and the way in which we solved the problem.

3. Parameter estimation minimizing the Gini index

3.1. Empirical distribution $\boldsymbol{\nu_1}$. We suppose to have a data set

$$P = \left\{ p_m = \left(p_1^{(m)}, p_2^{(m)}, \ldots, p_N^{(m)} \right) \right\}_{m=1}^{M},$$

with M elements in dimension N. Consider this set P as the following data frame:

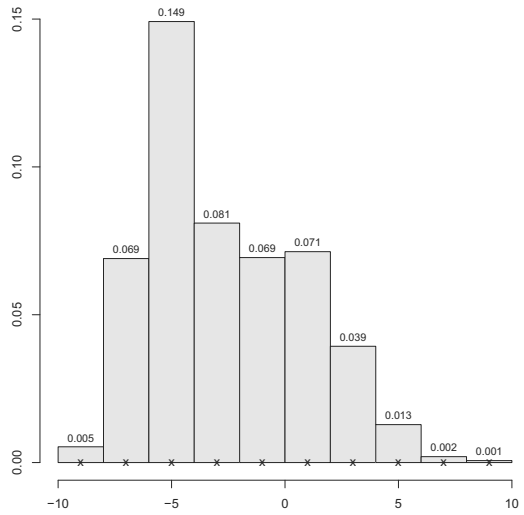


Fig. 1. Representative histogram for column C_n.

	C_1	C_2	...	C_N
p_1 =	$p_1^{(1)}$	$p_2^{(1)}$	...	$p_N^{(1)}$
p_2 =	$p_1^{(2)}$	$p_2^{(2)}$	...	$p_N^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
p_M =	$p_1^{(M)}$	$p_2^{(M)}$	...	$p_N^{(M)}$

For each $n = 1, \ldots, N$, we define data frame column sets as

$$C_n = \left\{ p_n^{(m)} \right\}_{m=1}^{M},$$

and we obtain a representative histogram of each of them. We assume that $C_n \in [\alpha_n, \beta_n]$. The histogram helps us to divide the set $[\alpha_n, \beta_n]$ into several bins. We use the count of these bins as a density estimate. If we have S_n bins, the partition is

$$B_1^n = [y_0^{(n)}, y_1^{(n)}), \quad B_2^n = [y_1^{(n)}, y_2^{(n)}), \quad \ldots, \quad B_{S_n}^n = [y_{S_n-1}^{(n)}, y_{S_n}^{(n)}].$$

The variable $Y_n = \left\{ y_j^{(n)} \right\}_{j=0}^{S_n}$ is represented by the x's in Fig. 1. In such a case, for a given point $x_n \in B_j^n$ we define the density estimation as

$$\hat{\gamma}_n(x_n) = \frac{1}{M} \sum_{m=1}^{M} I_{B_j^n}(p_n^{(m)}),$$

where I_A is the indicator function of the set A.

To define the empirical distribution ν_1 in X, we consider the multiplication of the density estimation for each of the columns, i.e., for $x = (x_1, x_2, \ldots, x_N) \in X$ we have

$$f_1(x) = \prod_{n=1}^{N} \hat{\gamma}_n(x_n). \tag{4}$$

3.2. Parametric distribution ν_2. For the parametric distribution ν_2 on X, we consider a Gaussian mixture density, that is,

$$f_2(x) = \sum_{k=1}^{K} \phi_k f(x|\mu_k, \Sigma_k).$$

The parameters ϕ_k are the mixture proportions and must comply with (2) and (3). In this work, the function f is assumed that is an *independent multivariate normal distribution* of dimension N, given in (1), where the mean μ_k is a real vector $[\mu_{k1}, \mu_{k2}, \ldots, \mu_{kN}]^T$ and the covariance matrix Σ_k is a real diagonal positive definite $N \times N$, matrix i.e.,

$$\Sigma_k = \text{diag}(\sigma_{k1}^2, \sigma_{k2}^2, \ldots, \sigma_{kN}^2). \tag{5}$$

Then, by the form of Σ_k, we know that

$$f(x|\mu_k, \Sigma_k) = \prod_{n=1}^{N} g(x_n|\mu_{kn}, \sigma_{kn}^2),$$

for $x = (x_1, x_2, \ldots, x_N) \in X$, where g is the univariate normal density function, that is,

$$g(s|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(s-\mu)^2}{2\sigma^2}}.$$

Thus, we have, for $x = (x_1, \ldots, x_N) \in X$,

$$f_2(x) = \sum_{k=1}^{K} \phi_k \cdot \prod_{n=1}^{N} g\left(x_n|\mu_{kn}, \sigma_{kn}^2\right). \tag{6}$$

Remark 1. It should be noted that we use independent probability densities in order not to increase the computational cost of the proposed model. This fundamental assumption is made to simplify the real-world problem complexity in a similar way to the naive Bayesian classification model (Mao *et al.*, 2020), and take individual problems for each coordinate in the multidimensional case as done by Kulczycki (2018). In several works (e.g., Elkan, 1997; Flach and Lachiche, 2004), it has been mentioned that even with this unrealistic assumption this technique is effective in practice.

3.3. Gini index problem for a Gaussian mixture model. The Gini index problem is as follows:

Minimize

$$\int_{X \times X} d(x, y)\, \mathrm{d}\pi$$

subject to

$$\pi(A, X) = \int_A f_1(x)\,\mathrm{d}x,$$
$$\pi(X, A) = \int_A f_2(x)\,\mathrm{d}x$$
$$\pi \in \mathcal{M}^+(X \times X),$$

for all A in the Borel σ-algebra of X, where $\mathcal{M}^+(X \times X)$ in the set of joint probability distributions on $X \times X$, d is the Euclidean distance in X, f_1 is as in (4) and f_2 is as in (6). We are looking for the following parameters of the density f_2 given in (6):

- the proportions defining the mixture $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_K)$, which comply with (2) and (3);
- the means of the Gaussian components $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_K)$;
- the covariance matrices of the independent Gaussian components $\boldsymbol{\Sigma} = (\Sigma_1, \Sigma_2, \ldots, \Sigma_K)$ as in (5).

3.4. Parameter estimation. As the solution exists, by Bassetti *et al.* (2006), we can solve this problem using the Lagrange multiplier method. The Lagrangian for this problem is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = &\sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{X}} d(x, y)\pi(x, y) \\ &- \sum_{x \in \boldsymbol{X}} \lambda_x \left[\sum_{y \in \boldsymbol{X}} \pi(x, y) - f_1(x) \right] \\ &- \sum_{y \in \boldsymbol{X}} \gamma_y \left[\sum_{x \in \boldsymbol{X}} \pi(x, y) - f_2(y) \right] \\ &- \alpha \left[\sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{X}} \pi(x, y) - 1 \right] \\ &- \beta \left[\sum_{k=1}^{K} \phi_k - 1 \right].\end{aligned} \tag{7}$$

Consider first setting the values for the joint distribution π. We write $\pi_{xy} = \pi(x, y)$ and $d_{xy} = d(x, y)$. For s and t fixed at X, the following is true:

$$\frac{\partial \mathcal{L}}{\partial \pi_{st}} = \mathrm{d}_{st} - \lambda_s - \gamma_t - \alpha = 0.$$

Then it is true that $d_{st} = \lambda_s + \gamma_t + \alpha$. Since d is a distance in X, if $s = t$, we have that $\gamma_t = -\lambda_t - \alpha$, due to $d_{tt} = 0$. Also, as $d_{ts} = d_{st}$, it is true that $\lambda_t + \gamma_s + \alpha = \lambda_s + \gamma_t + \alpha$, and then $\lambda_t - \lambda_s = -(\lambda_t - \lambda_s)$. This equality is fulfilled if and only if $\lambda_t = \lambda_s$, for $s, t \in X$.

Similarly, it is true that $\gamma_t = \gamma_s$ for $s, t \in X$. Then $\lambda_x = \boldsymbol{\lambda}$ for every $x \in X$, $\gamma_y = \boldsymbol{\gamma}$ for every $y \in X$ and $\boldsymbol{\gamma} = -\boldsymbol{\lambda} - \alpha$. With these equalities, the Lagrangian (7) turns out to be

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = &\sum_{x \in X} \sum_{y \in X} \mathrm{d}_{xy}\pi_{xy} + \boldsymbol{\lambda} \sum_{x \in X} f_1(x) \\ &- (\boldsymbol{\lambda} + \alpha) \sum_{y \in X} f_2(y) + \alpha - \beta \left[\sum_{k=1}^{K} \phi_k - 1 \right].\end{aligned} \tag{8}$$

To obtain the means, we differentiate the Lagrangian (8) with respect to μ_{tr}, with fixed $1 \le t \le K$ and fixed $1 \le r \le N$. Because the only sum in which the variable μ_{tr} appears is where ν_2 is present, we have

$$\frac{\partial \mathcal{L}}{\partial \mu_{tr}} = -(\boldsymbol{\lambda} + \alpha) \sum_{y \in X} \frac{\partial}{\partial \mu_{tr}} \nu_2(y)$$

In order to use the definition of f_2 given in (6), we must set the sum of all the elements of X given in the above equation in terms of their coordinates, so

$$\begin{aligned}&- (\boldsymbol{\lambda} + \alpha) \sum_{y \in X} \frac{\partial}{\partial \mu_{tr}} f_2(y) \\ &\quad = -(\boldsymbol{\lambda} + \alpha) \sum_{j_1=1}^{S_1} \sum_{j_2=1}^{S_2} \cdots \sum_{j_N=1}^{S_N} \frac{\partial}{\partial \mu_{tr}} \\ &\qquad \left[\sum_{k=1}^{K} \phi_k \cdot \prod_{n=1}^{N} g\left(y_{j_n}^{(n)} | \mu_{kn}, \sigma_{kn}^2 \right) \right].\end{aligned}$$

Thus, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mu_{tr}} = &-(\boldsymbol{\lambda} + \alpha) \cdot \phi_t \left(\frac{1}{\sigma_{tr}^2} \right) \\ &\cdot \sum_{j_1=1}^{S_1} \sum_{j_2=1}^{S_2} \cdots\cdots \sum_{j_N=1}^{S_N} \Big[\prod_{n=1}^{N} g\left(y_{j_n}^{(n)} | \mu_{tn}, \sigma_{tn}^2 \right) \\ &\cdot \left(y_{j_r}^{(r)} - \mu_{tr} \right) \Big] = 0.\end{aligned}$$

Here $\boldsymbol{\lambda} + \alpha$ must be different from 0, since if it were 0, $\boldsymbol{\gamma}$ would be 0 and there would be no restrictions. In the same way, ϕ_t must be different from 0, since if it were, there would be no K Gaussian components. Accordingly,

$$\begin{aligned}\sum_{j_1=1}^{S_1} \sum_{j_2=1}^{S_2} \cdots \sum_{j_N=1}^{S_N} \Big[&\prod_{n=1}^{N} g\left(y_{j_n}^{(n)} | \mu_{tn}, \sigma_{tn}^2 \right) \\ &\cdot \left(y_{j_r}^{(r)} - \mu_{tr} \right) \Big] = 0.\end{aligned}$$

Then we have

$$\mu_{tr} = \frac{\sum_{j_1=1}^{S_1} \sum_{j_2=1}^{S_2} \cdots \sum_{j_N=1}^{S_N} y_{j_r}^{(r)} \prod_{n=1}^{N} g\left(y_{j_n}^{(n)} | \mu_{tn}, \sigma_{tn}^2 \right)}{\sum_{j_1=1}^{S_1} \sum_{j_2=1}^{S_2} \cdots \sum_{j_N=1}^{S_N} \prod_{n=1}^{N} g\left(y_{j_n}^{(i)} | \mu_{tn}, \sigma_{tn}^2 \right)}.$$

Simplifying the common terms in the numerator and denominator, we get

$$\mu_{tr} = \frac{\sum_{j_r=1}^{S_r} y_{j_r}^{(r)} \exp\left(-\frac{(y_{j_r}^{(r)} - \mu_{tr})^2}{2\sigma_{tr}^2}\right)}{\sum_{j_r=1}^{S_r} \exp\left(-\frac{(y_{j_r}^{(r)} - \mu_{tr})^2}{2\sigma_{tr}^2}\right)}. \quad (9)$$

If we differentiate (8) with respect to σ_{tr}, with fixed $1 \leq t \leq K, 1 \leq r \leq N$, analogously to the previous case, we get

$$\sigma_{tr}^2 = \frac{\sum_{j_r=1}^{S_r} \left(y_{j_r}^{(r)} - \mu_{tr}\right)^2 \exp\left(-\frac{(y_{j_r}^{(r)} - \mu_{tr})^2}{2\sigma_{tr}^2}\right)}{\sum_{j_r=1}^{S_r} \exp\left(-\frac{(y_{j_r}^{(r)} - \mu_{tr})^2}{2\sigma_{tr}^2}\right)}. \quad (10)$$

The expressions (9) and (10) can be evaluated iteratively. This estimate is made with respect to the data set P, so the iterative expressions of the *GI algorithm* are

$$\mu_{tr} = \frac{\sum_{m=1}^{M} p_r^{(m)} \exp\left(-\frac{(p_r^{(m)} - \mu_{tr})^2}{2\sigma_{tr}^2}\right)}{\sum_{m=1}^{M} \exp\left(-\frac{(p_r^{(m)} - \mu_{tr})^2}{2\sigma_{tr}^2}\right)}, \quad (11)$$

for $1 \leq t \leq K$ and $1 \leq r \leq N$,

$$\sigma_{tr}^2 = \frac{\sum_{m=1}^{M} \left(p_r^{(m)} - \mu_{tr}\right)^2 \exp\left(-\frac{(p_r^{(m)} - \mu_{tr})^2}{2\sigma_{tr}^2}\right)}{\sum_{m=1}^{M} \exp\left(-\frac{(p_r^{(m)} - \mu_{tr})^2}{2\sigma_{tr}^2}\right)}, \quad (12)$$

for $1 \leq t \leq K$ and $1 \leq r \leq N$.

For the proportions of the mixture ϕ_k, as in the EM algorithm, we use in each iteration the expression (1) to calculate for each point p_m, $m = 1, \ldots, M$, the probability that p_m comes from the parametric distribution k, i.e., $P(f_k|p_m)$, with $k = 1, \ldots, K$, where $f_k = f(x|\mu_k, \Sigma_k)$. Then we can compare these values and determine the membership of the point p_m. Once this classification is made, we can obtain the proportions by taking the quantity of elements in each class and normalizing via dividing by M.

It is important to emphasize that, to obtain the expressions (11) and (12), we made the assumption that the components of the Gaussian mixture f_2 were independent multivariate normal distributions, and for this reason we only looked for the values of the covariance matrix that is a positive definite diagonal matrix. In the following section we will show some of the results obtained when using this algorithm.

4. Numerical results

In this section we will show the obtained numerical results when using the EM algorithm, the GI algorithm and the K-means method. We consider experiments with simulated data and real data. In the experiments with simulated data we consider data from 2, 3 and 4 Gaussians. For experiments with real data we consider two databases: the Iris data set and the Seeds data set, found in the UCI Machine Learning Repository [1].

To carry out these experiments we used the free software R[2]. For the classification with the K-means method we use the `kmeans()` function, and for the classification using the EM algorithm we employ the `GMM()` function in the `ClusterR` package.

With respect to the initial values that we use for the GI algorithm, we select them as follows. For the initial values of the means, we make a random selection of points within the domain of the analysed data set. For the initial values of the covariance matrix, we randomly select values between 1 and 4. For the initial values of the mixing proportions, we use, as usual, the values of a uniform distribution, that is, if we want adjust K Gaussians, we take proportions $\phi_k = 1/K$, for each k, as the initial values.

4.1. Simulated data. First we will show how the training data was generated and then what process will be used to make comparisons between the GI algorithm, the EM algorithm and the K-means model.

We generated a training data set P with a sample size $M = 3000$, from 2, 3 or 4 classes when considering different configurations for each case (*univariate* and *bivariate*), by analyzing the following characteristics:

- data proportion: the data amount from each class can be *equal* or *different*;
- data intersection: classes can be *spatially well differentiated* or *intersected*.

We specify the values used in each configuration in the corresponding sections. Once the data set to be analyzed has been generated, we translate the data items to the interval $[-10, 10]$ for the univariate case, and to $[-10, 10] \times [-10, 10]$ for the bivariate case. In addition, we can obtain a class vector that establishes from which Gaussian each data item comes from, in order to be able to establish the classification accuracy with the analysed models.

For the experiments with synthetic data we perform the following process 100 times:

1. We generate a set P of 3000 training data items specifying the characteristics of the configurations.

[1] `https://archive.ics.uci.edu/ml/index.php`.
[2] `https://www.r-project.org/`.

Table 1. Results of Configuration 1: univariate case.

Adjusted Gaussians	Acc.	GI algorithm	EM algorithm	K-means
2	Acc.	**100**	**100**	99.96667
	time	0.58	0.0345	0.0328
	GI	0.300322		
3	Acc.	**100**	78.8483333	75.66667
	time	0.5683	1.3175	0.2292
	GI	0.300322		
4	Acc.	**100**	77.0606667	74.3270
	time	0.6651	1.643	0.3069
	GI	0.300322		

Table 2. Results of Configuration 2: univariate case.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
3	Acc.	87.5	86.732	81.0850
	time	0.5689	0.0995	0.2305
	GI	1.408632		
4	Acc.	**99.899333**	94.36667	99.86667
	time	0.4542	1.6996	0.3029
	GI	1.109472		
5	Acc.	**99.899333**	83.776333	89.8310
	time	0.6337	2.0762	0.3678
	GI	1.109472		

2. We adjust each of the three models to the data set P, assuming that there are 2, 3, 4 or 5 groups, depending on each case. Then we obtain the classification accuracy.

3. We record the obtained classification accuracy through the three models and, for the univariate case, the Gini index values given by the GI algorithm.

Once we obtain the results of 100 iterations of these experiments, we calculate the average classification accuracy for each model; the time average, and the average values of the Gini index in the univariate case. These values are recorded in the tables in the following sections. In each table in this document, we show in boldface the best classification accuracy average. Let us recall that a small value for the Gini index suggests that the empirical distribution is closer to the estimated theoretical model.

4.1.1. Univariate case.

Configuration 1. (*See Fig. 2*(*a*)) We have 3000 data items generated with *equal proportions* of 2 *separate univariate* Gaussians, with the following parameters: 1500 data items with $\mu_1 = -12$, $\sigma_1 = 2$ and 1500 data items with $\mu_2 = 10$,$\sigma_2 = 3$. In this case we adjust 2, 3 and 4 Gaussians with the three methods, and the results are shown in Table 1.

The best classification accuracy average for the three models are those that consider the search for two Gaussians, because the data was generated precisely from two Gaussians. With the GI algorithm and the EM algorithm we obtain a 100% of average accuracy, higher than the K-means method. We obtain 100% when searching for 3 and 4 Gaussians with the GI algorithm because, even when it finds them, one or two have a proportion $\phi_k = 0$. For the value of the Gini index, we can see that the minimum is obtained for 2 Gaussians, which is an advantage, since we do not require *a priori* knowledge about the number of components in the mixture.

Configuration 2. (*See Fig. 2*(*b*)) We have 3000 data points generated with *different proportions* of 4 *intersected univariate* Gaussians, with the following parameters: 375 data points with $\mu_1 = -22$, $\sigma_1 = 3$, 750 data points with $\mu_2 = -8$, $\sigma_2 = 2$, 750 data points with $\mu_3 = 5$, $\sigma_3 = 2$ and 1125 data points with $\mu_4 = 15$, $\sigma_4 = 1$. In this case we adjust 3, 4 and 5 Gaussians. In Table 2 we show the results for this configuration.

In this configuration the GI algorithm yielded the best result when considering 4 and 5 Gaussians, and the value of the Gini index is lower in these cases. Again, we find a proportion $\phi_k = 0$ when we adjust 5 Gaussians with the GI algorithm.

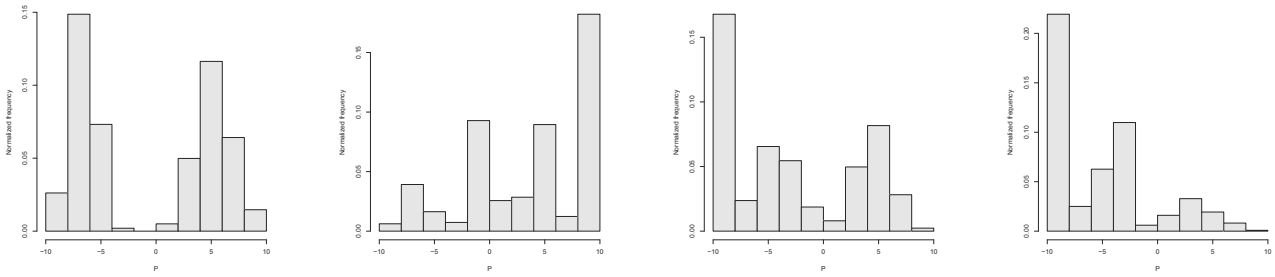
Configuration 3. (*See Fig. 2*(*c*)) We have 3000 data items generated with *equal proportions* of 3 *univariate* distributions, 2 *intersected*, with the following characteristics: 1000 data items from a beta distribution $X \sim \beta(2, 1.5)$, 1000 data items from a chi-square distribution $X \sim \chi^2(40)$ and 1000 data items from a Poisson distribution $X \sim Pois(100)$. We adjust 2, 3 and 4 Gaussians in this case. The results are shown in Table 3.

In this configuration we obtained a better classification when we adjusted 3 and 4 Gaussians with the GI algorithm. With this configuration we can see that, even when we generate the data set to be analyzed from non-Gaussian distributions, the method works correctly, that is, it is robust in this regard.

Configuration 4. (*See Fig. 2*(*d*)) We have 3000 data points generated with *different proportions* from 3 *intersected univariate* distributions, with the following characteristics: 1500 data points from a gamma distribution $X \sim \Gamma(0.7, 0.3)$, 1000 data points from a binomial distribution $X \sim Bin(20, 0.8)$, 500 data points from a Poisson distribution $X \sim Pois(35)$. The obtained results for this configuration are found in Table 4.

In this case, we obtained a better classification with the K-means method.

In Configurations 3 and 4 we obtain the same averages for 4 Gaussians with the GI algorithm, which



(a) Configuration 1 (b) Configuration 2 (c) Configuration 3 (d) Configuration 4

Fig. 2. Configuration examples for the univariate case.

Table 3. Results of Configuration 3: univariate case.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	66.666667	66.633	66.66667
	time	2.1092	0.1714	0.129
	GI	3.810997		
3	Acc.	**99.497333**	98.53267	94.7140
	time	3.3735	0.1525	0.8571
	GI	3.32223		
4	Acc.	**99.497333**	83.836333	76.66667
	time	3.5821	5.3871	1.1383
	GI	3.32223		

Table 4. Results of Configuration 4: univariate case.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	79.659	75.966667	81.63367
	time	0.3753	0.0779	0.033
	GI	1.987989		
3	Acc.	97.901333	97.501	**98.03333**
	time	1.0017	0.1584	0.2376
	GI	1.401458		
4	Acc.	97.901333	88.364667	87.26267
	time	1.055	0.4704	0.3044
	GI	1.401458		

means that in both cases the model found the same means and deviations values for 3 and 4 Gaussians, with a mixing proportion $\phi_k = 0$ for the case of 4 Gaussians. Although the results using the GI algorithm are not the best for Configuration 4, they are very close to those obtained with the K-means method, with the additional information of the covariance matrices.

These last two configurations have the ability to verify how the algorithm behaves with other distributions, not only for Gaussian mixtures. As can be seen, we obtain similar results to experiments with data generated from Gaussian mixtures.

In order to complement the experiments carried out in this section, we will analyze other situations that could be of interest, considering experiments with different amounts of data and different numbers of intervals in the representative histogram.

Different amount of data. In Tables 5 and 6, we show the results that we obtain by using the same parameters of Configurations 2 and 3, respectively, but varying the amount of data. In each table we show the data set size.

It is important to mention that the proportions used for the generation of the data correspond to the configuration proportions, i.e., the proportions $1/8$, $1/4$, $1/4$ and $3/8$ for Configuration 2 and a uniform distribution for Configuration 3.

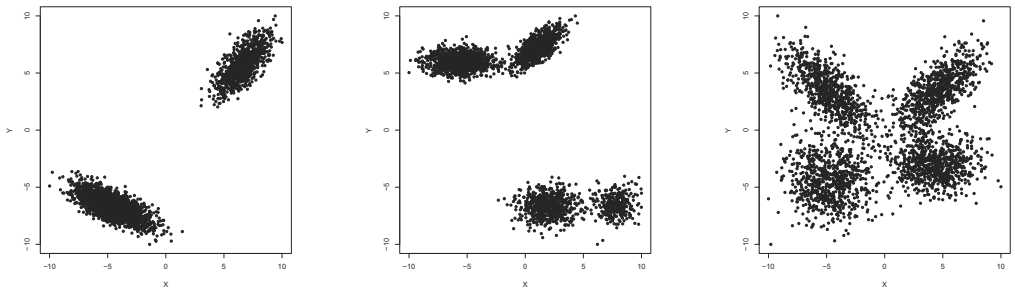
As you can see in Table 5 in addition to Table 2, the results given by the GI algorithm are not affected by the quantity of data. The values for the Gini index change in each case, but the minimum value always appears when we adjust 4 Gaussians. In Tables 3 and 6 the same thing is noticeable as in the previous case. It is important to note that, in this case, when we have less data, i.e., 150 elements, we obtain the best results. Again, we have the same accuracy for 3 and 4 Gaussians and the lowest Gini index in the same cases, because in the case of 4 Gaussians one of them has a zero proportion.

Different numbers of intervals in the representative histogram. For the experiments with different numbers of intervals in the representative histogram, we consider Configuration 2, with 4, 20 and 40 intervals. The results for this experiments are in Table 7. In Table 2, we show the results for 10 intervals.

As can be seen, the results do not depend on the number of intervals. The only value that varies with respect to the number of intervals is the Gini index. The lowest value appears when we adjust 4 Gaussians.

In all the experiments carried out in this work, we use the histogram generated by default by the `hist()` function of R, with 10 intervals.

4.1.2. Bivariate case. For the data generation in the plane, we consider independent bivariate normal distributions, that is, Gaussians whose covariance matrix



(a) Configuration 1 (b) Configuration 2 (c) Configuration 3

Fig. 3. Configuration examples for the bivariate case.

Table 5. Results of different amount of data for Configuration 2.

Set size	Adj. G		GI algorithm	EM algorithm	K-means
160	3	Ac.	86.875	86.5	86.875
		time	0.2451	0.0227	0.05
		GI	1.561819		
	4	Acc.	**99.875**	98.75	99.3750
		time	0.0794	0.0527	0.0674
		GI	1.184991		
	5	Acc.	99.375	88.125	84.3750
		time	0.1978	0.0715	0.0748
		GI	1.197076		
320	3	Ac.	87.5	85	86.86875
		time	0.4076	0.0412	0.14730
		GI	1.494086		
	4	Acc.	**99.75**	99.125	99.41875
		time	0.254	0.05430	0.1979
		GI	1.158949		
	5	Acc.	**99.75**	86.790625	84.10938
		time	0.2487	0.704	0.2149
		GI	1.15895		
1280	3	Ac.	85.12344	86.875781	84.99922
		time	1.45510	0.1826	0.5949
		GI	1.41227		
	4	Acc.	**99.898437**	99.207813	99.86766
		time	1.330626	3.2445	0.7271
		GI	1.130626		
	5	Acc.	99.4976563	82.293750	82.57734
		time	1.4255	4.1051	0.8474
		GI	1.1593782		

Table 6. Results of different amount of data for Configuration 3.

Set size	Adj. G		GI algorithm	EM algorithm	K-means
150	2	Ac.	66.666667	66.666667	66.66667
		time	0.0765	0.0195	0.0109
		GI	3.836915		
	3	Acc.	**99.98**	99.973333	97.30667
		time	0.079	0.0085	0.0288
		GI	3.371382		
	4	Acc.	**99.98**	87.42	84.70667
		time	0.0811	0.0231	0.0301
		GI	3.371382		
300	2	Ac.	66.666667	66.666667	66.66667
		time	0.17	0.0154	0.0094
		GI	4.026724		
	3	Acc.	**99.33667**	99.336667	98.00333
		time	0.1667	0.0115	0.0389
		GI	3.35724		
	4	Acc.	**99.33667**	87.226667	85.31333
		time	0.2154	0.0295	0.0538
		GI	3.35724		
1200	2	Ac.	66.66667	66.3325	66.66667
		time	0.2857	0.0365	0.0187
		GI	3.87596		
	3	Acc.	**99.9125**	99.078333	97.65833
		time	0.3832	0.0267	0.1359
		GI	3.376		
	4	Acc.	**99.9125**	87.972500	85.18583
		time	0.4729	0.6719	0.1533
		GI	3.376		

is diagonal, and non-independent bivariate normal distributions, that is, Gaussians whose covariance matrix is a positive definite non-diagonal matrix. This is in order to verify how efficient the models are when we use databases that might not meet the independence condition used by the GI algorithm.

Configuration 1. (*See Fig. 3*(*a*)) We have 3000 data items generated with *different proportions* of 2 *separate bivariate* Gaussians, with the following parameters: 2000 data items with

$$\mu_1 = (-7, -3), \quad \Sigma_1 = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$$

and 1000 data items with

$$\mu_2 = (3, 10), \quad \Sigma_2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

In Table 8 the results for this configuration when we adjust 2, 3 and 4 Gaussians are shown.

We obtain the best accuracy with the GI algorithm for 2, 3 and 4 Gaussians, and the EM algorithm and

the K-means method for 2 Gaussians. Once again, we obtained satisfactory results when adjusting 2 or more Gaussians with the GI algorithm because with this method we have values $\phi_k = 0$ for the extra components considered in the adjustment of the mixture. Also, in this configuration, we consider a generated data set from non-independent Gaussian distribution, which means that it does not comply with the independence condition, and we have favourable results.

Configuration 2. (*See Fig. 3(b)*) We have 3000 data points generated with *different proportions* of 4 bivariate Gaussians *intersected* by pairs, with the following parameters: 1200 data points with

$$\mu_1 = (0, 16), \quad \Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix},$$

900 data points with

$$\mu_2 = (7, 18), \quad \Sigma_2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix},$$

600 data points with

$$\mu_3 = (8, -5), \quad \Sigma_3 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

and 300 data points with

$$\mu_4 = (14, -5), \quad \Sigma_4 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

The results for this configuration, when we adjust 3, 4 and 5 Gaussians, are shown in Table 9.

Again, in this configuration we obtain the highest percentage for each model when we consider 4 Gaussians, and the GI algorithm has the highest average.

Configuration 3. (*See Fig. 3(c)*) We have 3000 data items generated with *equal proportions* from 4 *intersected bivariate* Gaussians, with the following parameters: 750 data items with

$$\mu_1 = (0, 16), \quad \Sigma_1 = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix},$$

750 data items with

$$\mu_2 = (5, 16), \quad \Sigma_2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix},$$

750 data items with

$$\mu_3 = (0, 10), \quad \Sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

and 750 data items with

$$\mu_4 = (5, 11), \quad \Sigma_4 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Table 7. Results of different number of intervals in the representative histogram.

No. of intervals	Adj. G	Acc.	GI algorithm	EM algorithm	K-means
4	3	Ac. time GI	87.5 0.4551 5.710026	86.73533 0.1198	86.799 0.2823
	4	Acc. time GI	**99.899667** 0.5139 4.710025	94.907333 1.8622	99.832 0.365
	5	Acc. time GI	**99.899667** 0.9875 4.710025	83.800333 2.3186	81.07567 0.4648
20	3	Ac. time GI	87.5 0.4884 0.9650311	86.733333 0.1288	86.8 0.3123
	4	Acc. time GI	**99.9** 0.5834 0.3244576	94.966667 1.9476	99.83333 0.4083
	5	Acc. time GI	**99.9** 1.0465 0.3244576	83.7666667 2.3792	81.06667 0.501
40	3	Ac. time GI	87.5 0.4737 0.8450669	86.73333 0.1386	86.8 0.2887
	4	Acc. time GI	**99.9** 0.5339 0.2116182	94.96667 1.89810	99.83333 0.3835
	5	Acc. time GI	**99.9** 0.9864 0.2116182	83.76667 2.3081	81.06667 0.468

Table 8. Results of Configuration 1: bivariate case.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc. time	**100** 0.9672	**100** 0.0228	**100** 0.0105
3	Acc. time	**100** 1.5312	83.90367 0.1307	67.99533 0.0064
4	Acc. time	**100** 1.6254	52.33167 0.2826	62.93767 0.0072

Table 9. Results of Configuration 2: bivariate case.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
3	Acc. time	89.8144 1.0006	89.6008 0.1135	89.3394 0.0165
4	Acc. time	**98.0784** 1.0633	94.8428 0.7247	90.0292 0.0166
5	Acc. time	97.5006 1.3915	79.3204 2.8278	77.0836 0.0197

In Table 10 the obtained results are displayed.

In this case, we obtain high averages for 4 and 5 adjusted Gaussians with the GI algorithm. We obtain the second best result when we consider 4 Gaussians with the K-means method. With this example we can see that

Table 10. Results of Configuration 3: bivariate case.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
3	Acc.	73.3720	73.5376	73.7796
	time	1.9184	0.8386	0.0518
4	Acc.	**97.6964**	93.4616	97.1192
	time	1.749	1.5722	0.058
5	Acc.	**97.6964**	82.5796	87.0752
	time	1.3066	4.7118	0.0648

Table 11. Results of different amount of data for Configuration 2: bivariate case.

Set size	Adj. G		GI algorithm	EM algorithm	K-means
150	3	Ac.	89.89333	89.34	89.34
		Time	0.2482	0.0091	0.0074
	4	Acc.	**99.99333**	99.34	99.34
		Time	0.1834	0.0109	0.0055
	5	Acc.	99.98667	85.3	85.20667
		Time	0.2792	0.0755	0.0051
300	3	Ac.	88.67667	89.66667	89.66667
		Time	0.2811	0.0141	0.0061
	4	Acc.	**99.99667**	80.00333	99.43
		Time	0.268	0.0428	0.0059
	5	Acc.	**99.99667**	69.9700	82.76
		Time	0.3704	0.0497	0.0058
1200	3	Ac.	89.66833	89.75167	69.915
		Time	0.7935	0.0242	0.0049
	4	Acc.	**99.91417**	99.75083	99.58083
		Time	0.8211	0.0444	0.0062
	5	Acc.	**99.91417**	87.90667	85.9925
		Time	1.1526	0.1197	0.0062

the method is robust with respect to the independence condition established for the analyzed data set.

In the same way as in the univariate case, we will show some experiments considering a different amount of data. In this case, we think that the experiments with different numbers of intervals are not relevant since we got similar results to those observed in Table 7. The number of intervals does not affect the results. Furthermore, we can see that the algorithm performs well even though the independence hypothesis is not fulfilled.

Different amounts of data. For the experiments with different amounts of data, we use the same parameters for the means and covariance matrices of Configurations 2 and 3 in the bivariate case. We show the different amounts of data and the results in Table 11 for Configuration 2 and in Table 12 for Configuration 3. The proportions used correspond to the configuration ones, as in the univariate case.

Table 12. Results of different amount of data for Configuration 3: bivariate case.

Set size	Adj. G		GI algorithm	EM algorithm	K-means
160	3	Ac.	73.2125	74.8875	74.9625
		time	0.3401	0.0108	0.0061
	4	Acc.	**99.33875**	99.28125	99.33125
		time	0.2771	0.0101	0.0055
	5	Acc.	**99.33875**	83.0125	88.70625
		time	0.3963	0.0137	0.0053
320	3	Ac.	68.73438	73.44688	73.44375
		time	0.4153	0.0102	0.0041
	4	Acc.	**95.96563**	95.89062	95.65625
		time	0.3818	0.0193	0.003
	5	Acc.	**95.96563**	83.8250	86.26562
		time	0.6307	0.0409	0.0043
1280	3	Ac.	62.73438	73.98438	73.82812
		time	1.273	0.0581	0.0088
	4	Acc.	**97.89062**	96.07812	97.8125
		time	0.9137	0.0941	0.0073
	5	Acc.	**97.89062**	89.21875	86.79688
		time	1.5352	0.1195	0.0084

Table 13. Results for 2 Gaussians in dimension 5.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	**100**	**100**	**100**
	time	5.6146	0.0969	0.0393
3	Acc.	99.85305	76.25725	75.833
	time	6.6641	8.4049	0.0642
4	Acc.	97.23465	67.4484	66.45755
	time	8.4456	12.0507	0.0843

4.1.3. Higher dimensions. In this section we consider experiments in dimensions 5 to 8 when we generate data from 2 and 3 Gaussians, in order to verify how efficient the models are in higher dimensions.

For this experiments we use the following values for the parameters: $\mu_1 = (1, 1, \ldots, 1) \in \mathbb{R}^N$ and $\Sigma_1 = \text{Diag}(1, 1, \ldots, 1)$ with size $N \times N$, $\mu_2 = (10, 10, \ldots, 10) \in \mathbb{R}^N$ and $\Sigma_2 = \Sigma_1$. Remember that N is the corresponding dimension of the analyzed data set. For the cases with 3 Gaussians, $\mu_3 = (20, 20, \ldots, 20) \in \mathbb{R}^N$ and $\Sigma_3 = \Sigma_1$.

In these cases we also transfer the data to

$$\underbrace{[-10, 10] \times \cdots \times [-10, 10]}_{N \text{ times}},$$

taking into account the number of dimensions N.

Dimension 5. In Tables 13 and 14, we show the results obtained when we generated the data set with 2 and 3 Gaussians, respectively.

In the first case we obtained 100% of accuracy when we adjusted 2 Gaussians with the 3 models. We obtain the

Table 14. Results for 3 Gaussians in dimension 5.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	66.66667	66.66667	66.66667
	time	5.508	0.212	0.074
3	Acc.	99.98667	**100**	**100**
	time	7.196	0.202	0.088
4	Acc.	96.16133	84.44	83.442
	time	8.92	21.052	0.118

Table 15. Results for 2 Gaussians in dimension 6.

Adjusted Gaussians		GI Algorithm	EM Algorithm	K-means
2	Acc.	**100**	**100**	**100**
	Time	0.7409	0.0115	0.0059
3	Acc.	**100**	77.5225	75.896
	Time	1.0445	0.9674	0.0054
4	Acc.	**100**	70.5875	68.2905
	Time	1.3269	1.3206	0.008

Table 16. Results for 3 Gaussians in dimension 6.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	66.66667	66.66667	66.66667
	time	0.638	0.033	0.015
3	Acc.	96.13667	93.68333	**100**
	time	0.833	0.875	0.033
4	Acc.	93.13333	60.35	83.86667
	time	1.086	1.216	0.036

second best result when we adjust 3 Gaussians with the GI algorithm. In the second scenario, we have better results with the EM algorithm and the K-means method when we adjust 3 Gaussians. We obtain the second best result with the GI algorithm.

It is important to mention that if we do not have *a priori* information about the number of classes in the data set and we adjust a wrong number of classes, we obtain the best results with the GI algorithm.

Dimension 6. We show the results for the experiments in dimension 6 in Tables 15 and 16.

For the experiments with 2 Gaussians, we obtain better results when we adjust 2 Gaussians with 3 models and when we adjust 3 and 4 Gaussians with the GI algorithm. In this case we can observe that the GI algorithm guarantees better results even when we adjust more Gaussians because the model finds values equal to 0 for mixture proportions for the additional Gaussians. As for 3 Gaussians, we obtain better results with the K-means method.

As in the experiments in dimension 5, in these cases we obtain the best result if we adjust a wrong number of classes with the GI algorithm.

Table 17. Results for 2 Gaussians in dimension 7.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	**100**	**100**	**100**
	time	1.0257	0.012	0.0064
3	Acc.	99.9975	76.7625	76.701
	time	1.3625	0.9881	0.0066
4	Acc.	99.9495	72.1115	68.392
	time	1.6538	1.4013	0.0087

Table 18. Results for 3 Gaussians in dimension 7.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	66.66667	66.66667	66.66667
	time	0.724	0.024	0.02
3	Acc.	90.71333	90.70667	**100**
	time	0.93	0.578	0.048
4	Acc.	89.53333	87.377333	84.01333
	time	1.212	0.75	0.028

Table 19. Results for 2 Gaussians in dimension 8.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	**100**	**100**	**100**
	time	0.862	0.03	0.016
3	Acc.	**100**	81.97	75.53
	time	1.232	0.798	0.032
4	Acc.	**100**	74.24	67.73
	time	1.536	1.368	0.032

Dimension 7. In Table 17 we can observe the results obtained when we generated a data set from 2 Gaussians. The best results are obtained when we adjust 2 Gaussians with 3 models. In this case, we obtained some mixture proportions with values close to zero, so that the accuracy for 3 and 4 Gaussians is close to 100%.

For the case of 3 Gaussians, we obtain the best result with the K-means method, as can be seen in Table 18.

Dimension 8. In Table 19 we show the results for 2 Gaussians. Again, we obtain the best results when we adjust 2, 3 and 4 Gaussians with the GI algorithm and when we adjust 2 Gaussians with the 3 models.

For 3 Gaussians, we obtain the results in Table 20. The K-means method yields better results when we adjust 3 Gaussians. It is important to mention that with this model we only have values for the means of the data. Again, in these cases, if we adjust a wrong number of classes, we have the best result with the GI algorithm.

Remarks. Our chief remarks are as follows.

- In most cases we get better results with the GI algorithm than with the other two models, and when we do not get the best result, we get the second best.
- We obtain the best averages when considering the actual number of classes used to generate the data.
- We get better results when classes are well differentiated, graphically speaking.
- The difference between the average with the GI algorithm for the original number of classes and the following ones is very small, because, when we consider more classes, this model finds the parameters for the additional ones with proportion ϕ_k equal or close to 0.
- The Gini index values for the univariate case are lower when we consider the actual number of data classes.
- The model appears to be robust with respect to the number of data items and the number of intervals in the representative histogram.
- If we analyze the execution time of these methods, it is clear that the fastest and relatively effective method is the K-means ones; however, with it we cannot find values for the covariance matrices.
- When we compare the accuracy of the GI algorithm with that of the EM algorithm, the former performs better but takes longer to execute.
- We can ensure that the GI algorithm finds values of mixing proportions equal or close to zero when we fit more Gaussians than the actual amount used to generate the data.

4.2. Real data. To carry out classification experiments with real data, we consider the Iris data set and the Seeds data set of the UCI Machine Learning Repository. For each of these data sets we use the GI algorithm, the EM algorithm and the K-means method for data classification.

Similar results of these experiments are reported in our earlier work (López-Lobato and Avendaño-Garrido, 2020); however, here the execution times for the three considered models are added.

4.2.1. Iris data set. In the Iris data set the authors examine 3 different varieties of the Iris plant: Iris Setosa, Iris Versicolour and Iris Virginica, 50 instances each, considering 4 physical characteristics of this plants: sepal length, sepal width, petal length and petal width. In this case we have a 4-dimensional data set with 3 classes. The results obtained for this database are shown in Table 21.

We obtain the best percentage of accuracy for the 3 models when we consider 3 clusters, because the database has 3 differentiated classes. We obtain a better percentage of accuracy with the GI algorithm for 3 Gaussians.

Table 20. Results for 3 Gaussians in dimension 8.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	66.66667	66.66667	66.66667
	time	0.8067	0.0367	0.0167
3	Acc.	99.7	58.388889	**100**
	time	1.1267	1.04333	0.0333
4	Acc.	96.68889	91.72222	83.97778
	time	1.4367	1.47	0.04

Table 21. Results for the Iris data set.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	66.66667	66.66667	66.66667
	time	0.11	2.39	0.02
3	Acc.	**97**	96.66667	88.66667
	time	0.22	3.36	0.01
4	Acc.	96	92	72.66667
	time	0.28	4.32	0.02

Table 22. Results for the Seeds data set.

Adjusted Gaussians		GI algorithm	EM algorithm	K-means
2	Acc.	66.66667	66.66667	66.19048
	time	1.07	2.68	0.01
3	Acc.	**94.61905**	93.33333	89.52381
	time	1.27	4.83	0.02
4	Acc.	92.85714	89.52381	77.14286
	time	1.7	9.97	0.02

4.2.2. Seeds data set. In the Seeds data set, the authors examine 3 different varieties of wheat seeds: Kama, Rosa and Canadian, 70 instances each, considering 7 geometrical parameters of wheat grains: area, perimeter, compactness, length of kernel, width of kernel, asymmetry coefficient and length of kernel groove. Thus, we have a 7-dimensional data set with 3 classes.

By adjusting this data set through the GI algorithm, the EM algorithm, and the K-means method, we obtain the percentages of accuracy shown in Table 22.

With the three models we obtain a better percentage of accuracy when we consider three adjusted Gaussians. We obtain the best average with the GI algorithm.

It is important to mention that, in the experiments with real data, the time required by the GI algorithm is smaller than that required by the EM algorithm with better results. This leads us to the conclusion that the proposed model helps us to efficiently estimate the parameters of a Gaussian mixture model through the Gini index problem.

5. Conclusions and future work

Thanks to the experiments carried out in this work, we can say that with the proposed model we obtain favourable

results, because our model seeks to minimize the Gini index between the empirical distribution and the proposed parametric distribution.

With the proposed model we obtain good results, even if the independence condition is not met and if the analyzed data sets comes from distributions that are not Gaussian mixtures, as in Configurations 3 and 4 for the univariate case and the experiments with real data. Furthermore, if we do not know the number of classes present in the database and arbitrarily set this number, the model fits the true number of classes, as we observed with the simulated data.

As future work, we want to study the theoretical properties and convergence of the GI algorithm and search for applications with real data. Also, we want to find some way to automatically specify the number of classes present in the analyzed data set with our method, analogously to the articles by Kulczycki (2018) and Kłopotek *et al.* (2020).

Acknowledgment

A.L. López Lobato acknowledges the support from the Mexican National Council for Science and Technology (CONACYT) through a national scholarship for PhD studies at the Faculty of Mathematics in the University of Veracruz.

References

Bassetti, F., Bodini, A. and Regazzini, E. (2006). On minimum Kantorovich distance estimators, *Statistics and Probability Letters* **76**(12): 1298–1302.

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*, Springer, New York.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society: Series B (Methodological)* **39**(1): 1–22.

Elkan, C. (1997). Boosting and naive Bayesian learning, *Proceedings of the International Conference on Knowledge Discovery and Data Mining, Newport Beach, USA*.

Flach, P.A. and Lachiche, N. (2004). Naive Bayesian classification of structured data, *Machine Learning* **57**(3): 233–269.

Giorgi, G.M. and Gigliarano, C. (2017). The Gini concentration index: A review of the inference literature, *Journal of Economic Surveys* **31**(4): 1130–1148.

Greenspan, H., Ruf, A. and Goldberger, J. (2006). Constrained Gaussian mixture model framework for automatic segmentation of MR brain images, *IEEE Transactions on Medical Imaging* **25**(9): 1233–1245.

Kłopotek, R., Kłopotek, M. and Wierzchoń, S. (2020). A feasible k-means kernel trick under non-Euclidean feature space, *International Journal of Applied Mathematics and Computer Science* **30**(4): 703–715, DOI: 10.34768/amcs-2020-0052.

Kulczycki, P. (2018). Kernel estimators for data analysis, *in* M. Ram and J.P. Davim (Eds), *Advanced Mathematical Techniques in Engineering Sciences,* CRC/Taylor & Francis, Boca Raton, pp. 177–202.

López-Lobato, A.L. and Avendaño-Garrido, M.L. (2020). Using the Gini index for a Gaussian mixture model, *in* L. Martínez-Villaseñor *et al.* (Eds), *Advances in Computational Intelligence. MICAI 2020*, Lecture Notes in Computer Science, Vol. 12469, Springer, Cham, pp. 403–418.

Mao, C., Lu, L. and Hu, B. (2020). Local probabilistic model for Bayesian classification: A generalized local classification model, *Applied Soft Computing* **93**: 106379.

Meng, X.-L. and Rubin, D.B. (1994). On the global and componentwise rates of convergence of the EM algorithm, *Linear Algebra and its Applications* **199**(Supp. 1): 413–425.

Povey, D., Burget, L., Agarwal, M., Akyazi, P., Kai, F., Ghoshal, A., Glembek, O., Goel, N., Karafiát, M., Rastrow, A., Rose, R., Schwarz, P. and Thomas, S. (2011). The subspace Gaussian mixture model: A structured model for speech recognition, *Computer Speech & Language* **25**(2): 404–439.

Rachev, S., Klebanov, L., Stoyanov, S. and Fabozzi, F. (2013). *The Methods of Distances in the Theory of Probability and Statistics*, Springer, New York, pp. 659–663.

Reynolds, D.A. (2009). Gaussian mixture models, *in* S.Z. Li (Ed.), *Encyclopedia of Biometrics*, Springer, New York, pp. 659–663.

Rubner, Y., Tomasi, C. and Guibas, L.J. (2000). The Earth mover's distance as a metric for image retrieval, *International Journal of Computer Vision* **40**(2): 99–121.

Singh, R., Pal, B.C. and Jabr, R.A. (2009). Statistical representation of distribution system loads using Gaussian mixture model, *IEEE Transactions on Power Systems* **25**(1): 29–37.

Torres-Carrasquillo, P.A., Reynolds, D.A. and Deller, J.R. (2002). Language identification using Gaussian mixture model tokenization, *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, USA*, pp. I–757.

Ultsch, A. and Lötsch, J. (2017). A data science based standardized Gini index as a Lorenz dominance preserving measure of the inequality of distributions, *PloS One* **12**(8): e0181572.

Vaida, F. (2005). Parameter convergence for EM and MM algorithms, *Statistica Sinica* **15**(2005): 831–840.

Villani, C. (2003). *Topics in Optimal Transportation*, American Mathematical Society, Providence.

Xu, L. and Jordan, M.I. (1996). On convergence properties of the EM algorithm for Gaussian mixtures, *Neural Computation* **8**(1): 129–151.

Adriana Laura López-Lobato is a third-year PhD student at the Faculty of Mathematics in the University of Veracruz. She was born in Xalapa, Mexico, in 1991. She received a BS degree and an MS degree, both in mathematics, at the University of Veracruz. Her scientific interests are applied mathematics, computing mathematics, probabilistic inference problems and optimization.

Martha Lorena Avendaño-Garrido was born in Xalapa, Mexico, in 1980. She received her BS degree in mathematics from the University of Veracruz (Mexico) in 2004, her MSc degree in industrial mathematics and computer science from the Mathematics Research Center (Mexico), and her PhD degree in mathematics from the Complutense University of Madrid (Spain) in 2013. She is a full professor at the Faculty of Mathematics of the University of Veracruz. She is interested in optimization, applied mathematics, modeling and computer science.

Received: 29 March 2021
Revised: 18 June 2021
Accepted: 1 July 2021