UNIVERSIDAD NACIONAL DE COLOMBIA

Parameter estimation in mixture models using evolutive algorithms

Natalia Romero Ríos

Universidad Nacional de Colombia
Facultad de Ciencias, Escuela de Estadística
Medellín, Colombia

2015

Parameter estimation in mixture models using evolutive algorithms

Natalia Romero Ríos

Thesis as a parcial requisite to obtain the title of:
MsC in Statistics
Director:
Ph.D. Juan Carlos Correa Morales

Universidad Nacional de Colombia
Facultad de Ciencias, Escuela de Estadística
Medellín, Colombia
2015

« It is the mark of a truly intelligent person to be moved by statistics.»
- George Bernard Shaw

Acknowledgments

I want to thank my mother, father, sister, husband and best friend for their continuos support, not only in this study but in my entire life.

I also thank Professor Juan Carlos Correa for his continuos guide of this research, and Professors Nelfi González and Juan Carlos Salazar for their recommendations for the improvement of this research.

Executive summary

The mixture models are widely used in cases when there are elements that come from diverse populations, mixed in a superpopulation. i.e. the proportions of expresed genes, and the weight of colombian $100 coins, year 1994. There are two main approaches for the modelling of mixture models: the bayesian and the clasical method. In the bayesian approach, the data are modelated and fitted to a given distribution, for example, the Dirichlet distribution. Further, the data are clustered for the posterior analysis. The classical method is the maximum likelihood estimation, using the Expectation-Maximization (EM) algorithm. This last method needs, as initial data, the amount of populations and their proportions in the superpopulation. Often, these data are very difficult to know or measure, because of the unknown nature of the problem. For that reason, in this work we propose the use of evolutive algorithms, such as genetic algorithms, simulated annealing and taboo search, to estimate the parameters of the mixture models. We propose an algorithm for the comparison of evolutive and traditional methods, and we illustrate the use of this algorithm with a real application. We found that the evolutive algorithms are a competitive option to estimate the parameters in mixture models in the cases when the populations in the mixture follows a gamma distribution, the weights of the populations in the mixture are even and the sample size is bigger than 100 items. For the mixture of normal distributions and the estimation of the number of populations in a mixture, the traditional method is a better option than the genetic algorithm.

Keywords: Mixture estimation, Statistics, Data analysis, Mixture data, Mixture estimation, Evolutive algorithms, Genetic algorithms.

Resumen ejecutivo

Los modelos de mezclas son ampliamente usados en casos donde se tienen elementos de poblaciones diversas, unidos en una super población. Como ejemplos de éstos se encuentran las proporciones de genes expresados y el peso de monedas de COP$100 del año 1994. Para su modelación se han utilizado enfoques bayesianos, donde se utiliza la modelación de los datos y el ajuste a distribuciones, por ejemplo, la Dirichlet para la agrupación de los datos y su posterior análisis. Otro enfoque es el clásico, el cual se basa en la estimación con máxima verosimilitud, usando el algoritmo EM (*Expectation - Maximization*). Éste último necesita como datos iniciales la cantidad de poblaciones existentes y sus proporciones, datos que en la vida aplicada muchas veces son desconocidos. Es por esto que se proponen los algoritmos evolutivos, como lo son los algoritmos genéticos, *simulated annealing* y búsqueda tabú como métodos que pueden servir para encontrar los parámetros de estimación de los modelos de mezclas. Para el desarrollo de este estudio se desarrolló un algoritmo para la comparación de métodos evolutivos y tradicionales y se incluye un ejemplo de aplicación. Se encontró que los algoritmos evolutivos son una opción competitiva para la estimación de parámetros en distri-

buciones de mezclas en los casos cuando las poblaciones en la mezcla siguen una distribución gamma, los pesos en las poblaciones son balanceados y el tamaño de muestra es mayor de 100 items. Para las mezclas de distribuciones normales y la estimación del número de poblaciones en una mezcla, el método tradicional es una mejor opción que el algoritmo genético.

Palabras claves: Estimación de mezclas, Estadística, Análisis de datos, Datos de mezclas, Algoritmos evolutivos, Algoritmos genéticos.

Index

1. Introduction

The mixture models are statistical representations of an overall distribution with two or more subpopulations. The main idea behind these models is to represent the heterogeneity of the data. [22] In real world, some examples are the contribution of distinct populations on a mixture of organisms for selection or breeding, or the weights of the COP$100 coins, manufactured on 1994, where it is observed that some coins are heavier than others.

For the estimation of the parameters in mixture models, two approaches are widely used: the bayesian approach and the classical approach. In the bayesian approach the data is collected, plotted, smoothed and then, a given distribution, as the Dirichlet distribution, is fitted to them. Further, the data are clustered and analysed. The classical approach uses maximum likelihood as the estimator, and uses the Expectation-Maximization (EM) algorithm to find all the unknown parameters. But, for that, a set of initial values must be given, such as the total number of subpopulations and the proportions in the overall distribution. More details can be found on Section 2. Those initial parameters are, often, difficult to find, because of the nature of the problem or the lack of previous data, and sometimes it can be suggested or recommended with the experts criteria [22], although as the method to find the estimators, in this project, we propose the evolutive algorithms as a method to solve this problem.

Evolutive algorithms are methods of stochastic search, that can work in very complex problems without the assumptions of the traditional methods, such as the continuity and the existence of derivatives. Some examples are Simulated Annealing, Taboo Search and Genetic Algorithms. The first one, Simulated Annealing, works as an analogy of the change of temperature of the materials under an annealing process. Taboo Search uses a structured method to find the maximum of a function avoiding local optima by imposing restrictions or "taboo' and searching on the entire parameter space. Finally, Genetic Algorithms uses biological concepts as evolution, crossbreeding and selection to find the maximum of a function.
This study is the comparison between traditional and evolutive methods for the estimation of the parameters in mixture models. This study uses the EM algorithm and Genetic algorithm to illustrate the traditional and evolutive methods respectly, both explained in Chapter 2. An algorithm is proposed, developed in R [26] and is described in Chapter 3. The simulation study and the results are described in Chapter 4 and an application to real data is evaluated in Chapter 5. Finally, the conclusions of this study can be found in Chapter 6.

2. Background

A mixture is defined as a collection of "data arising from two or more populations mixed in varying proportions"[22]. Some examples of mixtures are given by Reynolds and Templin, [29], that includes estimation of the relative contributions of distinct populations in a mixture of organisms, or the estimation of genes from source of parental populations, and estimate contributions from the diet of wild animals, changes in gene expressions [19], stellar populations [24], McCrea *et al.* used mixture models for analyzing the probabilities of annual survival for wild animals [21]. Another practical example, is the data collected in the Universidad Nacional de Colombia sede Medellin, the estimated density is showed in figure **2-1**, where we can see this distribution as a multimodal one, a main concentration on the center, but with a smaller concentration on the right tail. This, may indicate the presence of an non-homogeneous sample, or a mixture of samples. The mixtures have important applications, e.g., in pattern recognition, image processing, speech recognition, clasification and clustering, among others [14].
As mentioned by McLachlan and Basford, [22,], the data of a mixture model can consist of p atributes that are measured on each of n entities. The objetive is to perform a segmentation of these entities into g groups, so the entities within a group are homogeneous. The data can be represented as $x_1, \ldots, x_n$ where each x_j is a p-dimensional vector, arising from a superpopulation G, which is a mixture of a finite number of populations, g, denoted as $G_1, \ldots, G_g$ in some proportions $\pi_1, \ldots, \pi_g$, respectively, where:

$$\sum_{i=1}^{g} \pi_i = 1, \qquad \pi_i > 0, \qquad (i = 1, \ldots, g)$$

The probability density function (p.d.f) of x in G can be represented in the finite mixture form

$$f(x; \phi) = \sum_{i=1}^{g} \pi_i f_i(x; \theta) \tag{2-1}$$

where $f_i(x; \theta)$ is the p.d.f. of the G_i-th population, and θ denotes the vector of unknown parameters associated with the parametric forms adopted for the g densities. It is assumed that the vector $\phi = (\pi, \theta)'$ of unknown parameters belongs to some parameter space Ω

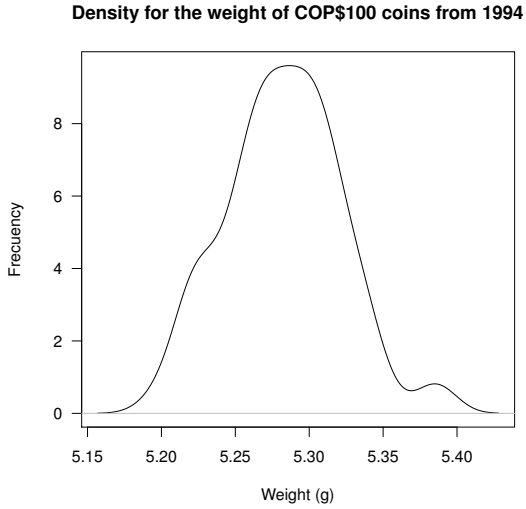

Figure 2-1.: Weights from year 1994 Colombian $100 coins. Source: Juan Carlos Correa from the Department of Statistics, National University of Colombia at. Medellín

2.1. Traditional method for the estimation of the parameters in mixture models

Some methods have been developed, to identify a mixture, including graphical, Bayesian approach, method of moments and the classical, maximum likelihood, approach [22].
The Bayesian approach incorporates prior information about the ϕ vector of parameters, taking these values from the criteria of the expert, labeling some observations as it is supposed to belong to a starting population [6]. The evaluation takes the form:

$$E(\theta|X_n) = \frac{E(\theta|X_{n-1})f(X_n|\theta)}{f(X_n|X_{n-1})}$$

The computation of the posterior unlabeled observation is difficult due to the form of the likelihood, because the number of terms grows exponentially with the sample size, n, and generally cannot be solved using analytical methods. Because of, Crawford *et al.* [6] proposed a modified Laplace method for the estimation of the parameters in a mixture model. One proposed technique is to take samples of the distribution and use a kernel technique to smooth the sample [38]. Another, as suggested by [19] is a nonparametric Bayesian approach, using a mixture of normal probabilities. In practice, the most used non-parametrical hierarchical mixture model, is the mixture of Dirichlet processes. With this model, a random discrete probability distribution is used as a mixing measure, and is the tool for modelling the clustering behavior [1]. For this, a non parametrical recursive estimator of the mixing

distribution was proposed by [25]. A problem with the Bayesian approach to estimate the parameters is the so called *label switching*, that is the "nonidentifiability of the components under symmetric priors". For this problem, Stephens proposed the relabelling algorithm [35]. For posteriors approximations, one technique consist in using response surface concepts, with high level polynomials [22]. Snee shows how this model can be reduced to have a good approximation to the results, making the entire model easier to understand and to compute. Also, it is proposed another technique, the ratio model, that can be used when "one or more of the components are limited to small ranges in the mixture"[34].
One of the most common approaches is the maximum likelihood estimation [22]. The likelihood estimation uses the EM algorithm (E for expectation, and M for maximation) of Dempster, Laird and Rubin [22]. To run this algorithm, it is needed some starting values for $\phi, \phi^{(0)}$ on the equation 2-1, or to initially partition the data into the specified number of groups g, and take $\phi^{(0)}$, as the estimate of ϕ based on this partition, as it represented the true grouping of the data [22].
To find and evaluate $\phi^{(0)}$ and ϕ, research has been conducted. For instance, Agha and Ibrahim [3] proposed an algorithm to find the Maximum Likelihood Estimation of Mixtures of Distributions., [29] developed a conditional likelihood ratio test of mixture homogeneity. [10] proposed a simultaneous estimation of the parameters under squared-error loss. [11] proposed a constrained nonparametric Maximum-Likelihood Estimation.,[14] proposed using a trimmed likelihood function to find the parameters when the mixture model has spurious outliers. [8] proposed an estimator of the number of populations that compose a mixture.
Those methods, for their requisites, as the initial values, are very difficult to apply into the real world. For that, in this work we suggest to use evolutive algorithms as a possible solution to find the value of the parameters in a mixture model. Some of these algorithms are described in the next section.

2.2. Optimization methods

When we want to optimize something, we want to make it better. This apply in every field of nature and human knowledge. To optimize something, we can use a function that describes the situation to optimize, and solve it using mathematical methods, or use heuristic methods. Some heuristic methods can be applied when an optimal solution is very complex to find, when the solution is sensible to some changes in the original data or when the function to be optimized does not fulfill the mathematical requisites as being continuous and have a known derivative, even when the function to be optimized is unknown [17] or when the available resources, such as computational time, are not sufficient to run a complete optimization [9]. A few examples of tools used by heuristic methods are: problem decomposing, that is dividing the problem into smaller spaces and find an optimal solution to each space, next step is to join the solutions together to find the general optima. Another method is using an inductive approach, based on generalization or taking the solution from a similar problem. Some other

method is to reduce the solution space, this can be done, for example by removing the points that do not satisfy the problem constraints or generate random solutions and selecting the best one [33]. The simplest heuristic algorithm, the so called *Exhaustive Search:* This method searches in the entire space. This guarantee to find the optimal solution, but it has the highest cost among the optimization methods [17], another is stochastic optimization, on which we use randomness in a constructive way. We look for an optimal x in the space S such as $f(x_{optim}) \leq f(x)$ for all $x \in S$. [13]. Another algorithm is the *Nelder-Mead Downhill Simplex Method:* This method create a simplex (the most elementary geometrical figure in a space of N dimensions), and then move it until it surrounds the minimum, and then contracting it until it is within an acceptable error. *Optimization Based on Line Minimization*, an algorithm begins at some random point on the surface, chooses a direction to move, then moves in that direction until the cost function begins to increase.

A problem with heuristic methods is that due to its simplicity, it might be stuck onto local optima. To overcome this issue, metaheuristics have been developed. According to [33] a metaheuristic is "An iterative master process that guides and modifies the operations of subordinate heuristics to produce efficiently high-quality solutions". Some metaheuristic methods include: Multilevel refinement, Beam search, Taboo search, Simulated annealing, Variable neighbourhood search, Guided local search, Multistart constructive approaches, Ant colony search, Simulated annealing and *Natural optimization*: These algorithms also head downhill from a starting point. These methods generate new points in the search space by applying operators to current points and statistically moving toward more optimal places in the search space. Some of these include: genetic algorithm, simulated annealing, particle swarm optimization, ant colony optimization, and evolutive algorithms [13].

2.2.1. Local Search

The local search (LS) algorithm, is a recurrent algorithm that starts with a given initial X_0 and it moves to an X_t given a $delta, \delta$ criteria [13] . It is the simplest optimization method, works as shown in figure **2-2** and it is known to have several disadvanges, such as:

- Stopping when it finds a local maximum
- Its results depend on the starting value X_0
- It does not have a stopping criteria, given by computational time. It means that the algorithm might run forever, if it does not find a local optima.

Because of those flaws, some improvements have been suggested by [13] such as methods that allow "jumping"when finding a local optima, and in order to search for several initials configurations or X_0

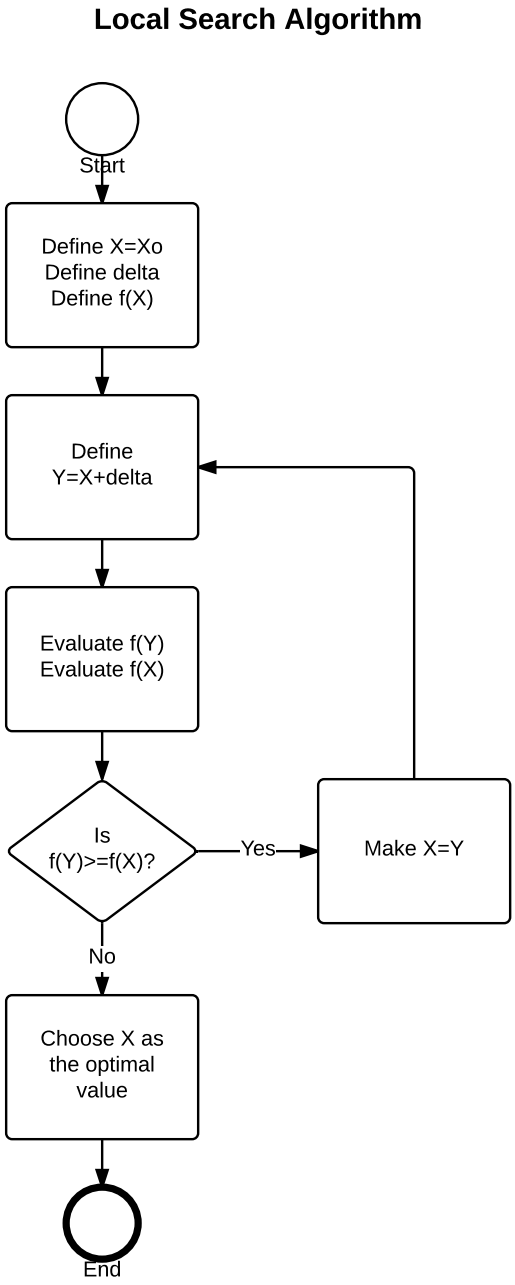

Figure 2-2.: Local Search Algorithm. Source: build by the authors

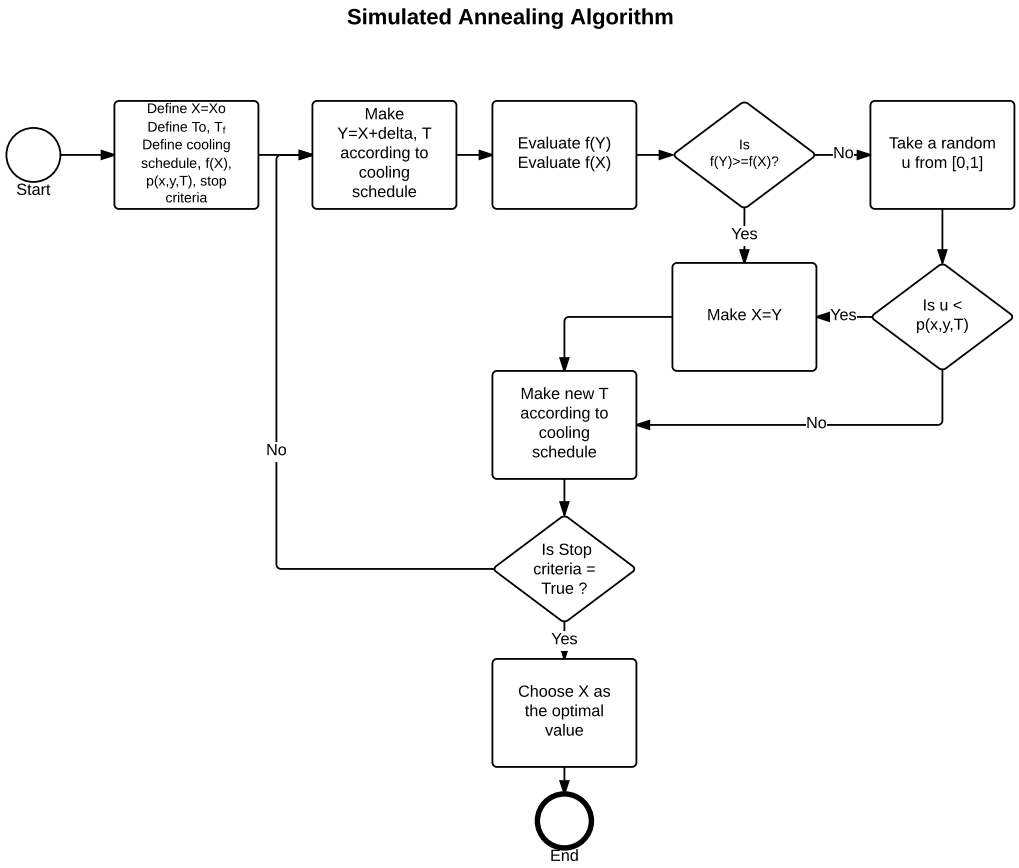

Figure 2-3.: Simulated Annealing Algorithm. [13]

2.2.2. Simulated Annealing

This computational algorithm, first proposed by Metropolis [23] uses an analogy of the transformation on a material configuration under a temperature change. This optimization method avoids getting stuck on local optima by searching on the neighbors and accepting "worse"moves, according to a given temperature and a cooling schedule, which controls the probability of jumping to worse moves. The algorithm is shown in figure **2-3**

For this basic algorithm Fouskakis and Draper [13] recommend some improvements, as to make temperature changes according to rejections or acceptances, rising it or making the temperature fall, respectively; to use an heuristic technique to find initial configurations, and making several parallel runs to further improvement.

2.2.3. Taboo Search

Taboo search is an heuristic method first proposed by Glover [15], and its goal is to scape from local optima. This algorithm also tries to avoid getting stuck on infinite loops by imposing restrictions, using an iterative framework divided into three steps: Preliminar search, intensification and diversification. In the first step, the algorithm begins in a point and searches into its neighbours for the maximal value into the objective function. When the algorithm can not find a neighbour with better performance, it moves to the neighbour with the best value, aside from the point where it stands, and forbids to go back by saving the previous move onto the taboo list. This step is done until a given number of iterations is reached. The next step is intensification, where this algorithm looks for a better solution, starting from the best move in the taboo list. This step is made a fixed number of iterations. Finally, in the diversification step, the algorithm looks for the most common moves marked as taboo on the taboo list, and looks into regions unexplored. This algorithm, as described by [13] is shown in figure **2-4**.
As noticed by [13], the taboo list size must be chosen very carefully, because if this value is very small, the algorithm might be stuck on a loop, however if this parameter is large, the search might have a big number of restrictions and, for that, it could not give a satisfactory result. For that, the authors suggest using a size of 7 or $\sqrt{p}$, being p the length of the string to be optimized.

2.2.4. Genetic Algorithms

Genetic algorithms (GA) are stochastic search models [40], first proposed by Holland (1975) in [13]. These models work as an analogy to Darwinian evolution, with their structural blocks, chromosomes, and making those evolve by selection, crossover and mutation [9]. The innovations proposed were "using bit string representations, proportional selection and crossover as the main operators"[32]. To implement a GA, first we must know the function to be optimized; later, a set of n chromosomes of length p are generated at random. The next step is to evaluate the fitness for every chromosome, and to arrange them by pairs, making the most fitted more likely to crossover, and there is a chance to their offspring to mutate. Later, only the most fitted between parents and their offspring are allowed to continue, and new chromosomes are generated. This is explained in figure **2-5**.
In general, the parameters for GA are: population size or n, number of iterations or generations t, crossover probability p_c, mutation probability p_m, and the objective and fitting function, $f(x)$ and $g(x)$, respectively, Czarn et. al. [7], found that crossover and mutation rates are statistical significant parameters, while the interaction within those two parameters is not, and that the best values for mutation rate lies from 0.0511 and 0.2092. [13] recommends the following configurations: $p_c > 0{,}3$, $p_m < 0{,}1$, and in some situations, they suggest using $p_c = 0{,}6$ and $p_m = 0{,}01$
The domain of the most optimization problems is $\mathbb{R}$, but often the GA look for the opti-

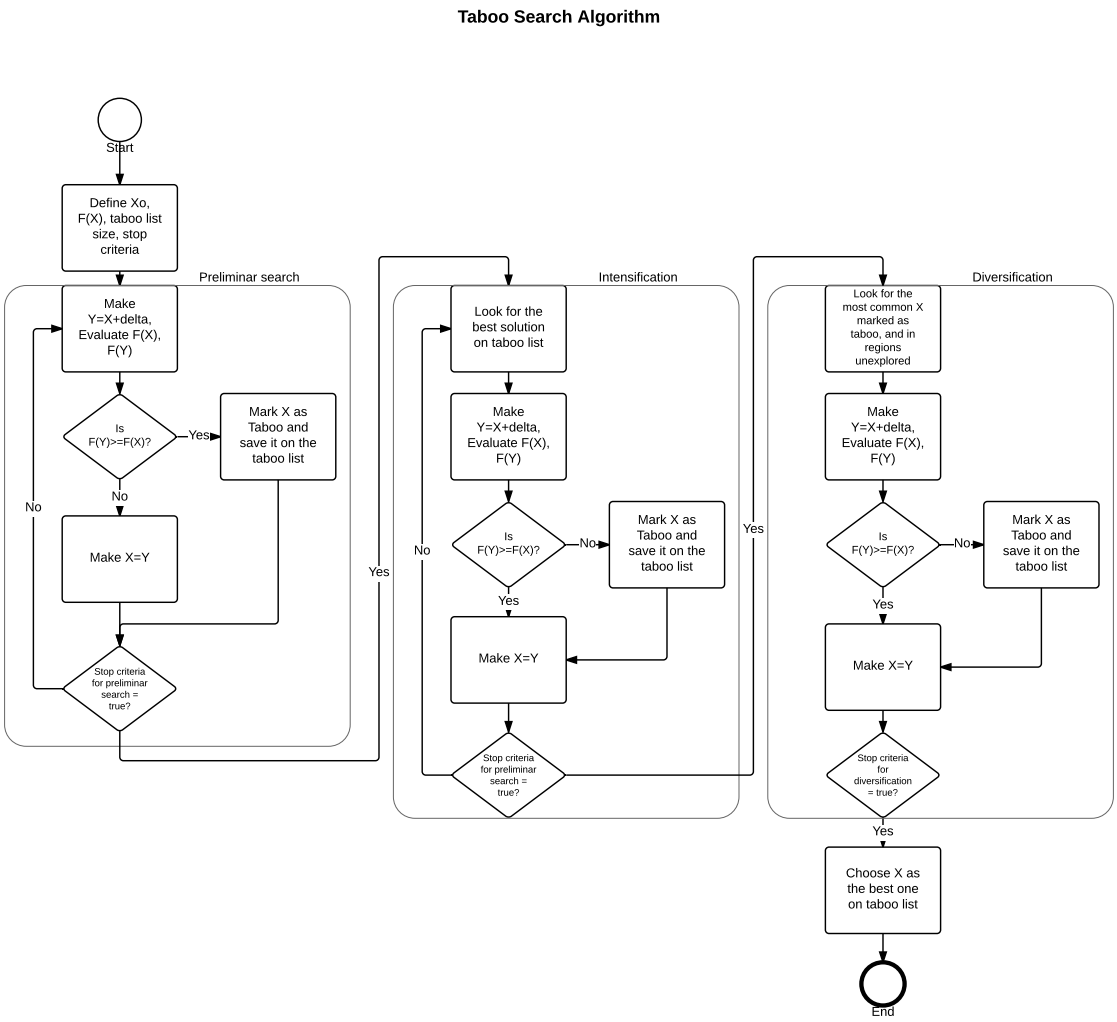

Figure 2-4.: Taboo Search Algorithm. Source: build by the authors

mization of binary integers. [30] suggests to represent the real valued solution as a binary sequence, and then mapping that onto the real numbers again. For that, if we were looking for a solution onto the space $[-d, d]$, then, for $-d$, we can use the binary sequence of length D, 00..,000, and 11..,111 will be d. Adding a binary 1 to a previous number, increases its value by $\frac{d}{2}^{D-1}$.
Although, it has been explained the basic GA, there are several modifications that can be included, as those proposed by Fouskakis and Draper [13], such as altering the selection mechanism for the most fitted chromosomes, adjusting the value from the fitting function when the children are worst fitted than the parents, and change the crossover operations.
In order to work, GA have less requisites than other mathematical optimization methods, for example: strict continuity, differentiability and convexity. That is the reason why, in general, it can be said that the results obtained by GA are weaker, although very good, compared to those obtained by mathematical methods [17]. However, when traditional (mathematical) approaches fail, GA can perform better, this is because GA takes a fixed point on the surface as a potential solution (chromosome), that with mating (crossover) and mutation can explore different points on the same space, and those points being evaluated by the fitness functions makes the fitness ratio increase for every generation, making the solution closer to the real one [30]. But, GA iskm 1—j intensive on the use of computational time and the parameters must be chosen carefully, to avoid getting stuck on a local optima, specially if the function is very sinuous, when the function has a lot of local maximum, or when the function have a big local maximum [30]

2.2.5. Applications in Statistics

Some application of Evolutive Algorithms to problems in statistics will be discused in this section, for example, [30] makes comparison of the performance of GA in a deterministic problem with no solution using mathematical approach, a problem of multiple regression, the estimation of the parameters in a logistic regression and finally, they estimate the parameters of a linear model using a robust criteria. Another approach to solve statistical problems via genetic algorithms was proposed by [36], who uses GA for outlier detection and variable selection in linear regression models, concluding that GA are a good approach for this type of situations, avoiding potential difficulties of smearing and masking.
In R [26], some packages have been developed to use Genetic Algoritms, In [32] some packages are listed in the appendix such as **gafit**, **galts**, **mcga**, **rgenoud**, **genalg** and **DEoptim** and a new one is proposed, **GA**. A review of these packages can be seen in Appendix A
For the purpose of this research, simulated data will be used for the experiments, following the steps proposed by Santner et. al (2003) [31]:

1. To identify the data to collect.

2. To design the experiment.

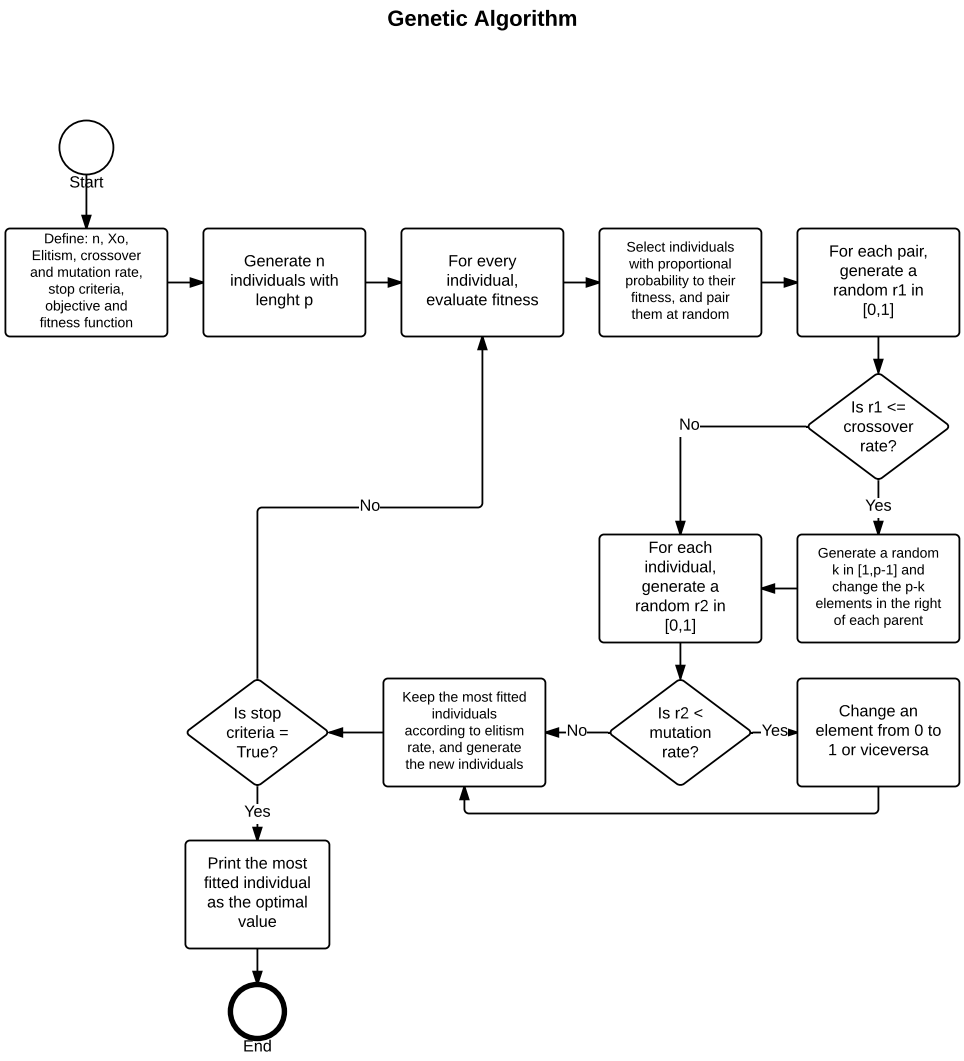

Figure 2-5.: Genetic Algorithm. Source: build by the authors

3. To execute the experiment for the final step that is to analyze the resulting data.

A comparison between the results of metaheuristic methods will be made, to evaluate their performance against the optimal solution or some benchmark and evaluate the computational requirements to find the solution [33]. When comparing an heuristic method to another, it is common to evaluate fitness against time to find the best run, but, as noticed by [20], more rigorous methods can be applied. For Genetic Algorithms, they propose comparing the medians with non-parametric tests or use statistical inference based on ranks, because parametric methods as ANOVA or t-student need the sample to be evaluated is a sequence of identically distributed values, and also, the data has to be independent, a requisite that is not fulfilled because of the evolutionary method, where a generation depends of the one before them.

3. Algorithm

The objective of this chapter is to explain the algorithm developed for this research, to achieve a full understanding of how the data were collected and analyzed, and why the conclusions of this research are valid.
First at all, in this project the factors used for running the simulations are:

- Distribution of the data
- Number of populations in the mixture
- Number of populations in the mixture known or unknown
- Sample size
- Separation of the distributions: We chose to test the distance between the populations to compare how can the methods detect a mixture when the distance between the means or shape parameters in densities, for the mixture of normals and gamma, respectively, exceed two or more standard deviations. The distance is measured between the means, for the mixture of normal distributions, and the shape parameter in mixtures of gamma distributions. [28]
- Populations weight on the sample

Some model assumptions for the simulations in this work:

- The distributions came from populations with the same variance
- The EM algorithm was used to illustrate traditional methods
- Genetic Algorithm was used to illustrate the evolutive algorithms
- The initial values for the EM algorithm were fixed as the real value + random number

Two generic algorithms were used in this work. Both were made using `R` [26]. One was used for the known number of distributions and it is described on figure **3-1** on page 15, because we can compute the error on the estimation for every parameter, and another when the number of populations was unknown, Figure **3-2** on page 18, because it needs a different approach.

3.1. Algorithm for when the number of populations is known

The algorithm for when the number of populations is known is described in figure **3-1**. The tasks on each activity are as follows:

1. **Define starting parameters:** This algorithm needs some parameters to start. These parameter are
   - k number of populations
   - π vector of the population weights for the k populations, such as $\sum_{i=1}^{k} \pi_i = 1$
   - $\theta_{i=1\ldots k}$vector of the parameters for each population. For the normal distribution, for example $\theta_i = (\mu_i, \sigma_i)$
   - Number of iterations. In this work it is used 1000 iteration for each simulation because is a common number in simulations experiments
   - Set the seed to control the random number generation
   - n Sample size of the data to be evaluated

2. **Create the data for the simulations:** For the simulations the data were created following the next steps:
   - Set population size. The number of data in the population i, n_i, is created at random, where $\sum n_i = n$, for every simulation. The number of data in each population n_i follows a multinomial distribution as $Multinom(n, \pi_i \ldots k-1)$ where n is the size of the sample, k is the number of populations in the mixture, and π is the vector of the weights
   - Generate n_i random data from the population with parameters θ_i. These data are the mixture of data

3. **Set starting values for the traditional algorithm:** The EM algorithm needs some initial values for the π_i and θ_i. For the simulations, the initial values are set as the $\theta_i + random\Delta$.

4. For the unknown number of populations **Set the initial number of populations**. This task is accomplished using the Gap Analysis proposed by Tibshirani (2000) [27]. For this task the package `lga` [16] is used.

5. **Estimate the parameters using the traditional method:** This estimation is made using the library `mixtools` [4] from the software `R` [26], to compute the Expectation maximization method, described in Section 2.2 in page 2.2

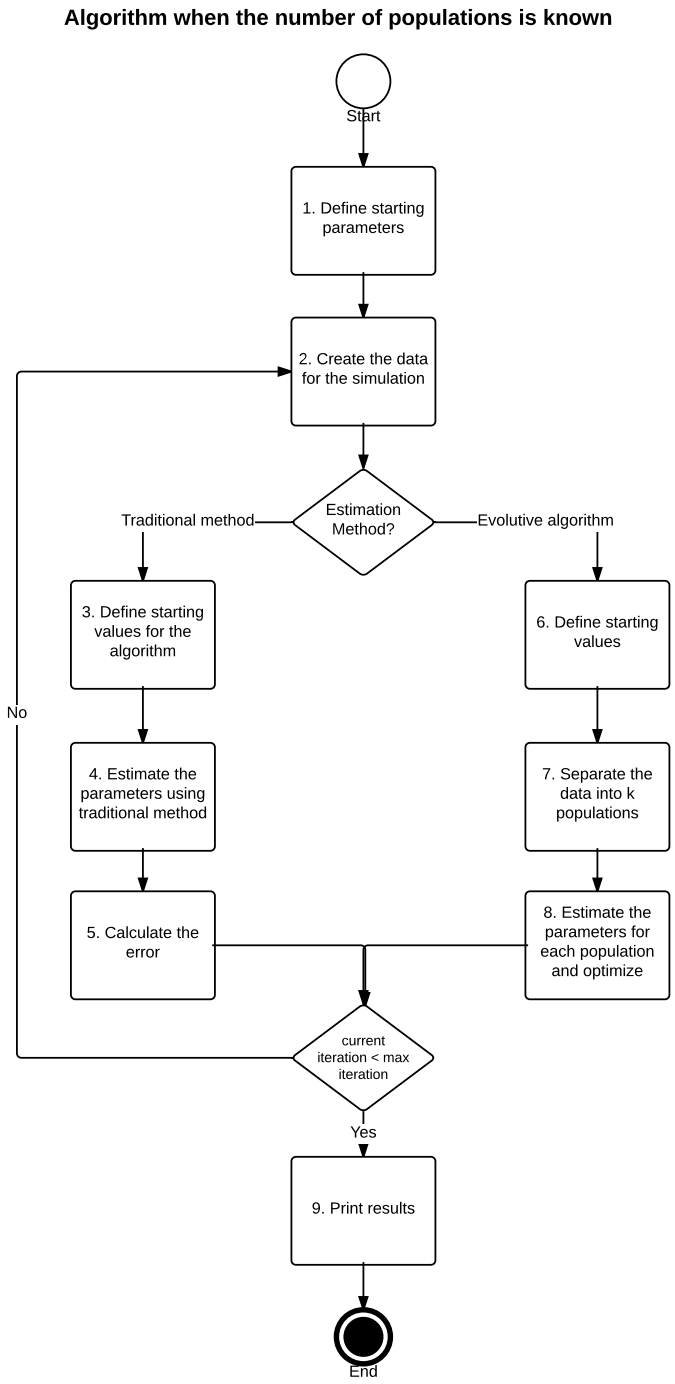

Figure 3-1.: Algorithm when the number of populations is known. Source: build by the authors

6. **Calculate the error:** The estimation of the error. This algorithm use a percentage $error = \frac{|\theta - \hat{\theta}|}{\theta} * 100\,\%$

7. **Define starting values:** For the evolutive method, genetic algorithm, the following parameters needs to be set:
   - Chromosome length. This value is set as n
   - Minimum and maximum values for the chromosomes: These values are set as 0 and k-1, because the populations are separated according to this value
   - Mutation chance. In this work a mutation chance of 0.05 is fixed, as sugested by [13]

8. **Segregate the data into k populations:** The data are segregated into k populations according to the value of the chromosome

9. **Estimate the parameters for each population and optimize:** The algorithm estimates the parameters using a maximum likelihood fitting for every population. This task is made using the library `MASS`. [37]. The next step is to maximize the likelihood using genetic algorithms, this is accomplishes using the library `genalg` [39]. The estimators are the following for the mixture of normal populations:

   $$\hat{\mu} = \frac{1}{n}\sum_{j=1}^{n} x_j$$

   and

   $$\hat{\sigma^2} = \frac{1}{n}\sum_{j=1}^{n}(x_j - \hat{\mu})^2$$

   and for the mixture of gamma populations:

   $$\hat{\theta} = \frac{1}{kN}\sum_{i=1}^{N} x_i$$

   and

   $$\psi(k) = \frac{\Gamma'(k)}{\Gamma(k)}$$

   the digamma function, that needs to be solved numerically.

10. **Print results:** The results are shown as graphics for the estimation of every parameter. Also, the results of every iteration is stored for posterior analysis

The last step is to compare the results of both methods, EM and GA. To compute the distance between the estimated and the real density, the Hellinger distance (HD) is used as an approach to measure the distance between the true and estimated densities, the one with the true parameters used for the simulation, $f(x)$, and the one with the parameters given by the EM and GA, $g(x)$ [5]. This is shown in equation 3-1. This estimator has been used before in mixtures of parametric families, as described by [18] and [2] and it has been shown that this estimator is robust. To analyze the result, when the distance is zero, it means that the estimated values are the same as the real ones, for this reason, the best method is the one to achieve the minimum values [2].

$$HD = \int_{-\infty}^{\infty} \left(\sqrt{f(x)} - \sqrt{g(x)}\right)^2 dx \approx \sum_{i=1}^{M} \left(\sqrt{f(x_i)} - \sqrt{g(x_i)}\right)^2 \tag{3-1}$$

where:

- X is a variable created to estimate the approximated distance expressed in Equation 3-1 in the interval $X \in I$, $I = \{(\mu_1 - 3\sigma_1); (\mu_k + 3\sigma_k))\}$, for the normal mixture, and $I = \{(\alpha_1 - 3\beta_1); (\alpha_k + 3\beta_k))\}$ for the gamma mixture. x_i, $i = 1..,500$, being $M = 500$, estimated in a grid of X, and the population k is the population with the mean or the scale parameter more
- $f(x)$ is the real density, calculed with the parameters used to generate the data in the simulation
- $g(x)$ is the estimated density, calculed with the parameters obtained with the EM or the GA

3.2. Algorithm when the number of populations is unknown

The algorithm when the number of population is unknown is described in Figure **3-2**.

The tasks on each step are as follows, there are several changes compared to the algorithm when the number of populations is known:

1. **Define starting parameters:** This algorithm needs some parameters to be set. These parameter are
   - k number of populations
   - π vector of the population weights for the k populations, restraint to $\sum_{i=1}^{k} \pi_i = 1$

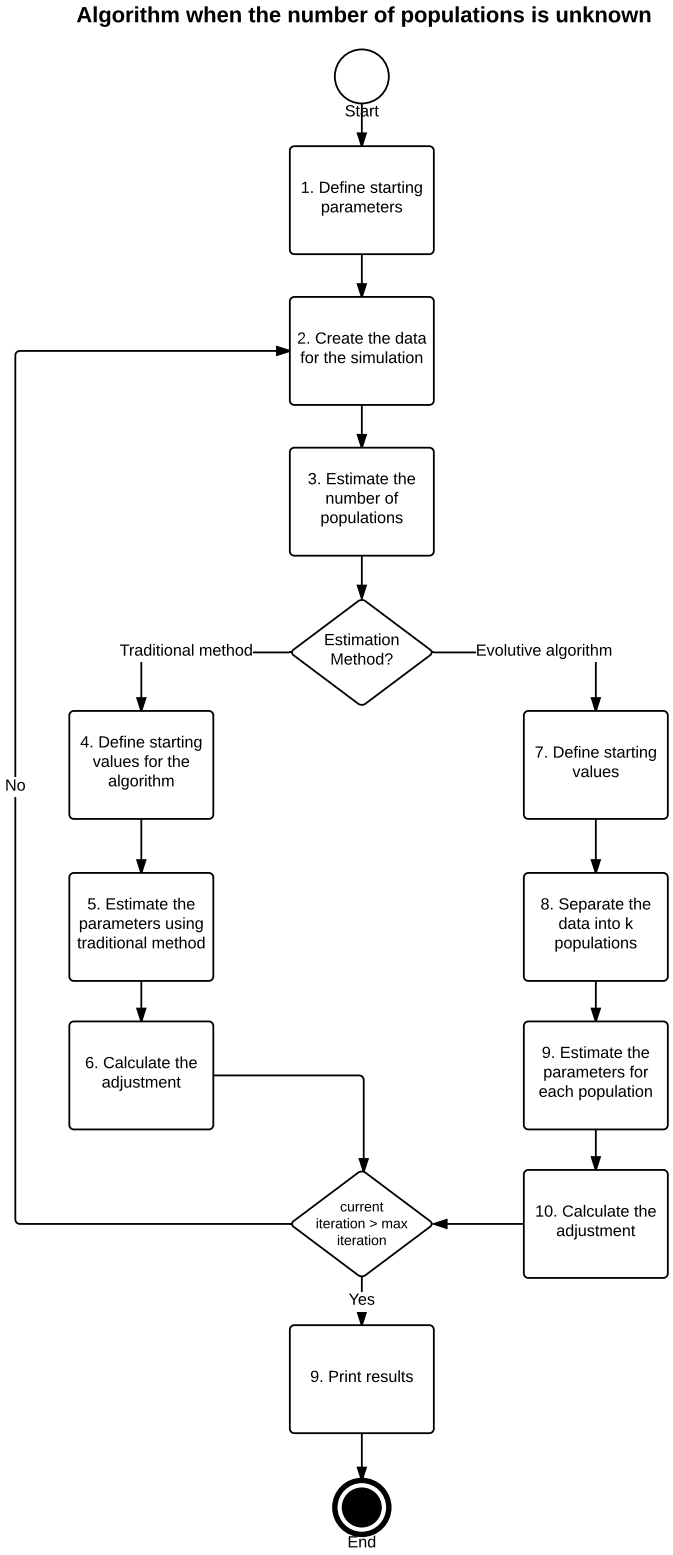

Figure 3-2.: Algorithm when the number of populations is unknown. Source: Build by the authors

- θ_i with $i = 1, ..., k$ vector of parameters for each population. For the normal distribution, for example $\theta_i = (\mu_i, \sigma_i)$
- Number of iterations. In this work are used 1000 iteration for every simulation because is a common number in simulations experiments
- Seed to control the random number generation
- n Sample size of the data to be evaluated

2. **Generate the data for the simulation** For the simulations the data was generated with the following steps:
   - Generate the size of population. The number of data in each population n_i follows a multinomial distribution as $Multinom(n, \pi_i \dots k-1)$ where n is the size of the sample and π is the vector of the weights
   - Generate n_i random data from the population with parameters θ_i. This data is the mixture of data
3. Estimate the number of populations For this step, the Gap statistic proposed by Tibshirani et. al. (2001) [27], and the library `lga` [16] were used for the estimation of k. This method estimates the number of populations or clusters using an iterative algorithm, that estimate the number of populations and then çomparing the change in within the cluster dispersion with the expected one under a reference distribution"
4. **Set starting values for the traditional algorithm:** The EM algorithm needs some initial values for the number of populations k, vector of the weights π_i and vector of the parameters θ_i. For the simulations, the initial values are set as the $\theta_i + random\Delta$
5. **Estimate the parameters using traditional method:** This estimation is made using the library `mixtools` [4], using the same estimators described in Section 3.1
6. **Calculate the adjustment:** Because the real number of populations and the calculated one may differ, we compare the average number of populations estimated with the real value.
7. **Define starting values:** For the evolutive method, genetic algorithm, the following parameters needs to be set:
   - Chromosome length. This value is set as n
   - Minimum and maximum values for the chromosomes: These values are set as 0 and k+2, because the populations are separated according to this value, and because the nature of the genetic algorithm, we can chose a wider range to evaluate the population

- Mutation chance. In this work a mutation chance of 0.05 was fixed, as sugested by Fouskakis and Draper (1996) [13]

8. **Segregate the data into k populations:** The data are segregated into k populations according to the value of the chromosome.
9. **Estimate the parameters for each population:** The algorithm estimates the parameters using a maximum likelihood fitting for every population, assuming the same distribution for each population in the mixture.
10. **Calculate the adjustment and evaluate the distance, and optimize:** The joint likelihood is estimated as the sum of the k' likelihoods, and for that sum and the vector of the parameters ($\pi_{1\ldots k'}; \theta_{1'_k}$ being k' the final number of populations according to the algorithm
11. **Print results:** The results are shown as graphics for the Hellinger distance among methods [5], and the times that the algorithms underestimate and overestimate the number of populations. Also, the results of every iteration is stored for posterior analysis

This chapter has the general schema of the algorithms developed for this study. The complete algorithms can be seen on the Appendix B. In the next chapter, the results of the simulations and the respective analysis are shown.

4. Simulation study

To make the comparation between traditional methods and evolutive algorithms to estimate the parameters in mixture models, we will implement a simulation study. This allow us to know the real parameters and find the error of the estimation. The basic form for mixture functions follow the equation 2-1 from the page 2. As an example, we are going to use a mixture of two normal distributions with parameters $\theta_1 = (\mu_1, \sigma_1)$ and $\theta_2 = (\mu_2, \sigma_2)$, respectively so the equation 2-1 follows the form:

$$f(x; \mu_1, \sigma_1, \mu_2, \sigma_2, \pi) = \pi * f_1(x, \mu_1, \sigma_1) + (1 - \pi) * f_2(x, \mu_2, \sigma_2)$$

In this case, the values to estimate are:

- μ_1, σ_1 Parameters of the first population
- μ_2, σ_2 Parameters of the second population
- π Population weights

For the case with the mixture of three gamma distribution, the equation 2-1 follows the form:

$$\begin{aligned} f(x; \alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3, \pi) &= \pi_1 * \frac{\beta_1^{\alpha_1}}{\Gamma(\alpha_1)} x^{\alpha_1 - 1} e^{-\beta_1 x} + \\ & \quad \pi_2 * \frac{\beta_2^{\alpha_2}}{\Gamma(\alpha_2)} x^{\alpha_2 - 1} e^{-\beta_2 x} + \\ & \quad (1 - \pi_1 - \pi_2) * \frac{\beta_3^{\alpha_3}}{\Gamma(\alpha_3)} x^{\alpha_3 - 1} e^{-\beta_3 x} \end{aligned}$$

In this case, the values to estimate are:

- α_i, β_i Parameters of the population i, with, in this case, $i = 1, 2, 3$
- π Vector of population weights

Factor	**Levels**					
Populations	Known	Unknown				
Mixture of distributions	Normal	Gamma				
k number of populations	2	3	5			
π_i population weights	5	10	25	50		
sample size	30	50	100	200	500	1000

Table 4-1.: Proposed values for the parameters of the simulation study. Source: Build by the authors

The initial plan to run the simulations was to test the parameters described on Table**4-1**.

This algorithm took around to 80 hours to compute a single experiment, with 1000 iterations. Because of that, to compute the complete run of the simulations might took around 17280 hours, or 720 days straight.

To adress this problem, three approaches were taken:

1. Run multiple machines. For this research project, we had six machines running the experiment 24 hours, 7 days straight, but this could not solve the time to finish on time. Also, we explore the possibility to buy virtual machines online, but that was more expensive that we could afford, due to the available budget for this thesis.

2. Run the algorithm on a server, but this did not improve significantly the computation time.

3. Change the package `genalg` [39] for `GA` [32], but this package takes three times longer to compute the genetic algorithm.

Finally, we decided to reduce the number of iterations of each experiment to 500, and to use a sample size of 30, 50 and 200. This modification allowed the experiment to be run on time for this research. The final configuration is shown in Table**4-2**

The Tables in the following sections of this chapter show the average distance $\overline{HD}$ and their standard deviation for every experiment.

Factor	Levels			
Populations	Known	Unknown		
Mixture of distributions	Normal	Gamma		
k number of populations	2	3	5	
π_i population weights	5	10	25	50
sample size	30	50	100	200

Table 4-2.: Final values for the parameters of the simulation study. Source: The authors

4.1. Evaluation of estimation of the parameters in mixtures, with known number of populations

The goal of this section is to evaluate the effect of the number of populations between the estimation of the parameters of mixture models when the number of populations is known, for evolutive algorithms and traditional methods. The columns have the following name codes:

- **Separation**: Is the distance between the means of the populations in the mixture, in the mixture of normal distributions, or the α parameter in the mixture of gammas, and it is measured in standard deviation and β parameter, respectively. All the populations in the mixture have a standard deviation and β equal to 1.
- **Weight**: Is the percentage weight of the first distribution, the weight of the other distributions in the mixture were asigned in equal parts to complete the 100 % and it can be seen in the column. For example, in a case with a mixture of three normal populations, in Table **4-5** in page 29, the fourth column has a weight of 5, and that is the weight shown in Figure **4-4**, the next column shows 2x47.5, because of the weight vector has the values of 5 %, 47.5 % and 47.5 %.
- **GA**: Genetic Algorithm
- **EM**: Expectation Maximization Algorithm
- **Mean**: Is the average between the Hellinger distance of the iterations with the respective configuration.
- **SD**: Is the standard deviation between the Hellinger distance of the simulations

For each one we had the means and the standard deviation from the 500 iterations for each experiment.

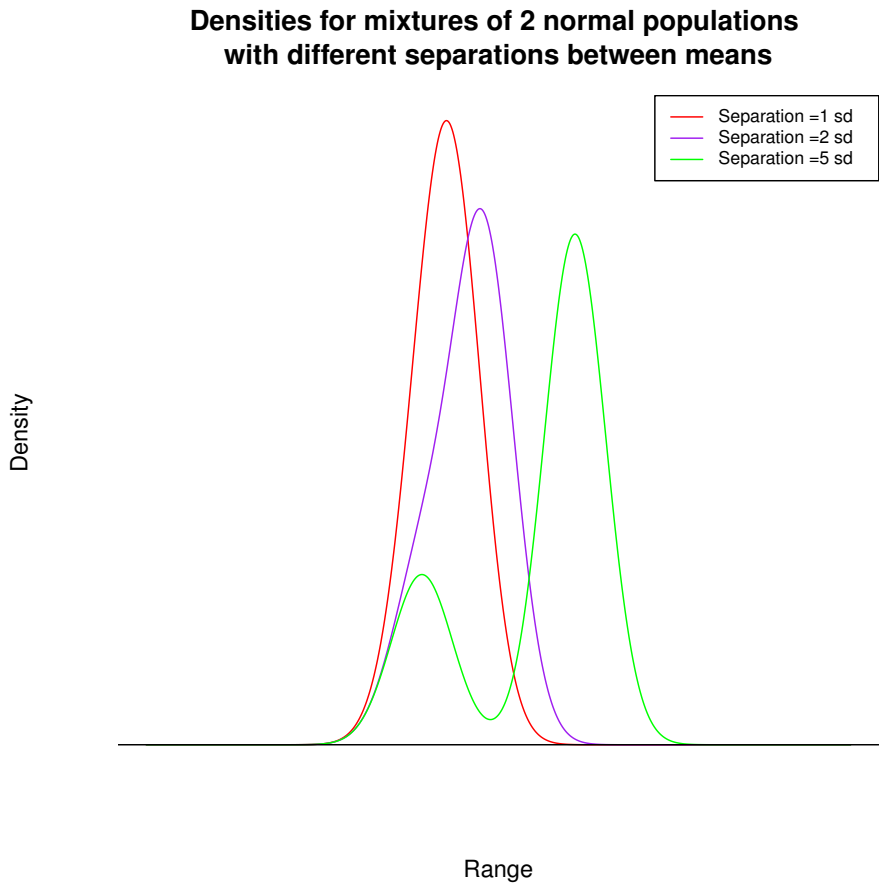

Figure 4-1.: Plot of the densities of a mixture of two normal distributions, with weight of 25 % and 75 %. Source: build by the authors

4.1.1. Mixture of two normal distributions

For the mixture of two normal distributions created in this study, an example of the densities can be seen in Figure **4-1**, where the representations of a mixture with composition 25 % and 75 % of each population, for different separations between means.

In Tables **4-3**, **4-4**, and in Figure **4-2**, it is shown the mean of the Hellinger distance estimated for the 500 iterations of the estimation of the parameters in a mixture of two normal populations. For this scenario, the separation between means increased the distance for the GA, with a more noticeable effect when the weights of the populations were uneven, below 50 %, but for the EM the separation between means had the opposite effect. The sample size had less impact on the results of the distance for the GA, because of the cases when the sample size was 100 and 200 items, and the weight was 10 %, the distance was similar. For the EM algorithm, as the separation increases and the weights of the populations were even, the mean HD decreases. For that behaviour, the EM seems to be a better alternative for

Hellinger Distance for a mixture of two populations, Part 1 of 2.									
	Parameters					Mean		SD	
#	Sample Size	Separation	Weight			GA	EM	GA	EM
1	30	1	5	95		NA	7.3719059	NA	5.0801785
2	50	1	5	95		NA	3.9671332	NA	2.6632592
3	100	1	5	95		0.7276752	1.7939734	0.7166283	1.3264134
4	200	1	5	95		0.3486274	0.794333	0.3158048	0.5639108
5	30	2	5	95		NA	6.3952341	NA	4.0574232
6	50	2	5	95		NA	3.6501297	NA	2.4086476
7	100	2	5	95		0.7223326	1.7279633	0.7971845	1.239071
8	200	2	5	95		0.6245537	0.8000868	0.320689	0.6049102
9	30	5	5	95		NA	6.9884434	NA	3.8216841
10	50	5	5	95		12.8977553	3.7425842	5.7476426	2.7131607
11	100	5	5	95		8.4840123	1.7340373	2.660486	1.562029
12	200	5	5	95		7.1815266	0.7393026	1.4068978	0.788641
13	30	1	10	90		NA	7.3162403	NA	5.0166785
14	50	1	10	90		1.5470258	3.9139956	1.5799118	2.767204
15	100	1	10	90		0.7274471	1.8512988	0.7548557	1.3020375
16	200	1	10	90		0.3555792	0.7972737	0.3609759	0.5848706
17	30	2	10	90		NA	6.6389644	NA	4.6860836
18	50	2	10	90		1.1922491	3.5581831	1.2855913	2.4344436
19	100	2	10	90		0.8183982	1.6432441	0.6932028	1.1947833
20	200	2	10	90		0.9489221	0.8090102	0.3203636	0.6187133
21	30	5	10	90		6.9927596	6.4536987	5.2863691	4.5691423
22	50	5	10	90		11.2706251	3.4488844	4.303969	3.0707152
23	100	5	10	90		12.2237259	1.5019521	1.3098497	1.5226873
24	200	5	10	90		11.1575835	0.6397216	0.8028084	0.488359

Table 4-3.: Hellinger estimated distance for a mixture of two normal populations and the number of populations is known, part 1. continues in Table**4-4** in page 26. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

the estimation of the parameters in a mixture of two normal distributions, because it could estimate the distance in all the cases, including the ones with small sample size and weight of one population below 10 %. The GA only had better performance that the EM when the weight was 50 % and the sample size was small.

4.1.2. Mixture of three normal distributions

For the mixture of three normal distributions created in this study, an example of the densities can be seen in Figure **4-3**, where the representations of a mixture with composition 25 %, 37.5 % and 37.5 % of each population, for different separations between means.
For the mixture of 3 normal population, with known number of populations, the mean

Hellinger Distance for a mixture of two populations, Part 2 of 2.								
	Parameters				Mean		SD	
#	Sample Size	Separation	Weight		GA	EM	GA	EM
25	30	1	25	75	2.309	6.9230101	2.3033108	5.0169856
26	50	1	25	75	1.4628363	3.8887844	1.5848417	2.9550561
27	100	1	25	75	0.7068634	1.6552048	0.6465543	1.1961168
28	200	1	25	75	0.366506	0.8190172	0.3979012	0.6090011
29	30	2	25	75	2.408683	6.4599304	2.3535815	3.9974919
30	50	2	25	75	1.3339603	3.7103984	1.282645	2.5387932
31	100	2	25	75	0.6877073	1.6371665	0.6714559	1.1295445
32	200	2	25	75	0.8038557	0.7948953	0.4860565	0.5828737
33	30	5	25	75	8.6900822	5.5200456	5.0504729	4.303661
34	50	5	25	75	10.7969615	3.0145286	3.4208612	2.6661729
35	100	5	25	75	10.4488628	1.3342895	1.5840213	1.4277715
36	200	5	25	75	9.2736561	0.5706032	0.410276	0.3843375
37	30	1	50	50	2.4265982	6.9557724	2.6363092	4.6296586
38	50	1	50	50	1.4585161	4.0261444	1.5101946	2.8861504
39	100	1	50	50	0.7616498	1.8145096	0.7925469	1.3100404
40	200	1	50	50	0.3638134	0.8333842	0.3394218	0.6350888
41	30	2	50	50	2.567045	6.5595071	2.6265313	4.322705
42	50	2	50	50	1.3765336	3.5292087	1.4553041	2.4209196
43	100	2	50	50	0.6848512	1.6761035	0.6560538	1.1150232
44	200	2	50	50	0.3198601	0.7889517	0.3205622	0.5014049
45	30	5	50	50	2.9445942	5.1580724	2.2080052	4.0815618
46	50	5	50	50	1.6267193	3.0855921	1.4252764	2.9185279
47	100	5	50	50	0.6848512	1.6761035	0.6560538	1.1150232
48	200	5	50	50	0.3198601	0.7889517	0.3205622	0.5014049

Table 4-4.: Hellinger estimated distance for a mixture of two normal populations and the number of populations is known, part 2. Is the continuation of the Table**4-3** in page 25.The NA is when the data could not be computed for the algorithm. Source: Build by the authors

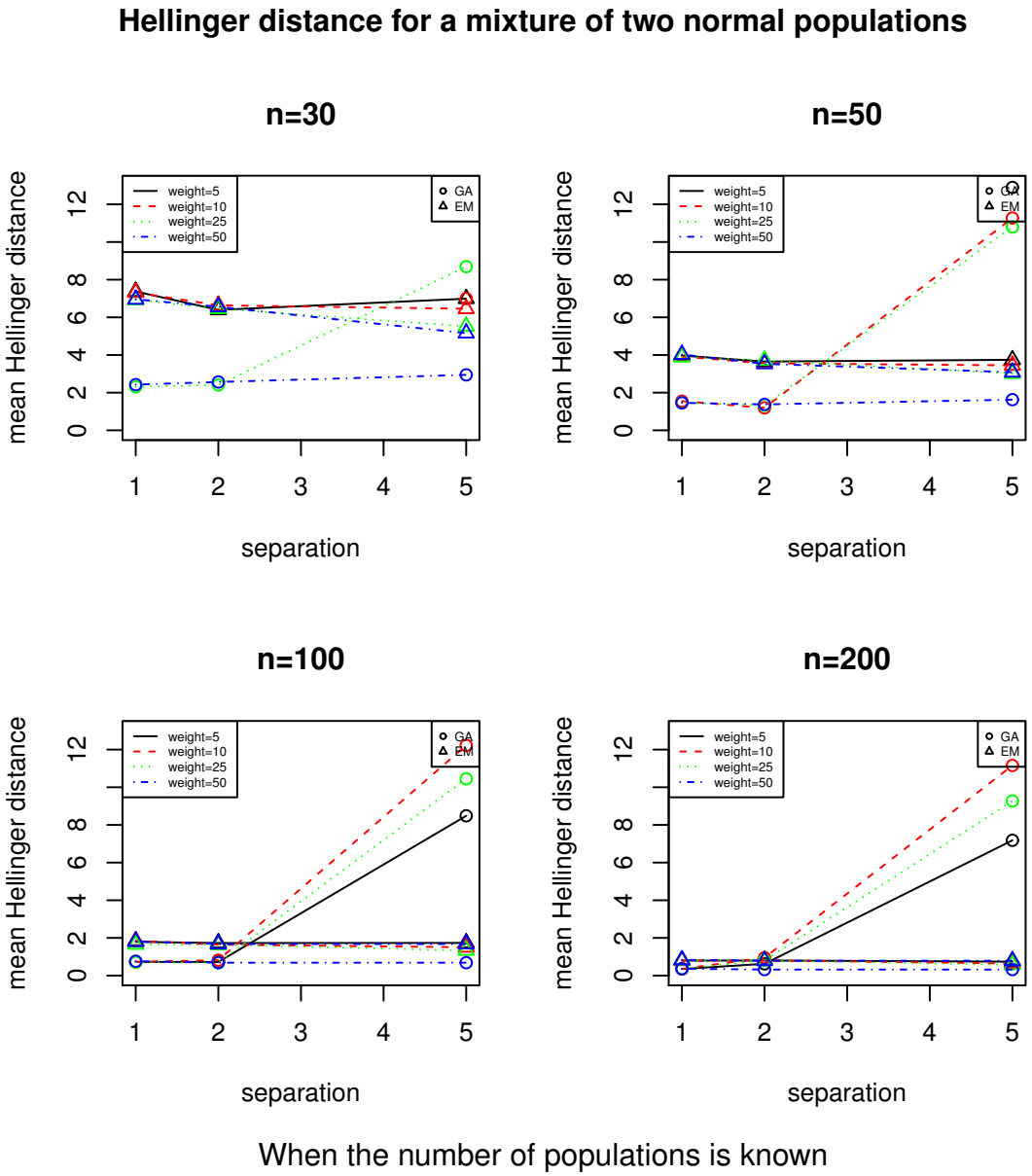

Figure 4-2.: Plot of the Hellinger distance for the mixture of 2 normal populations, in a mixture with known number of populations. Source: build by the authors

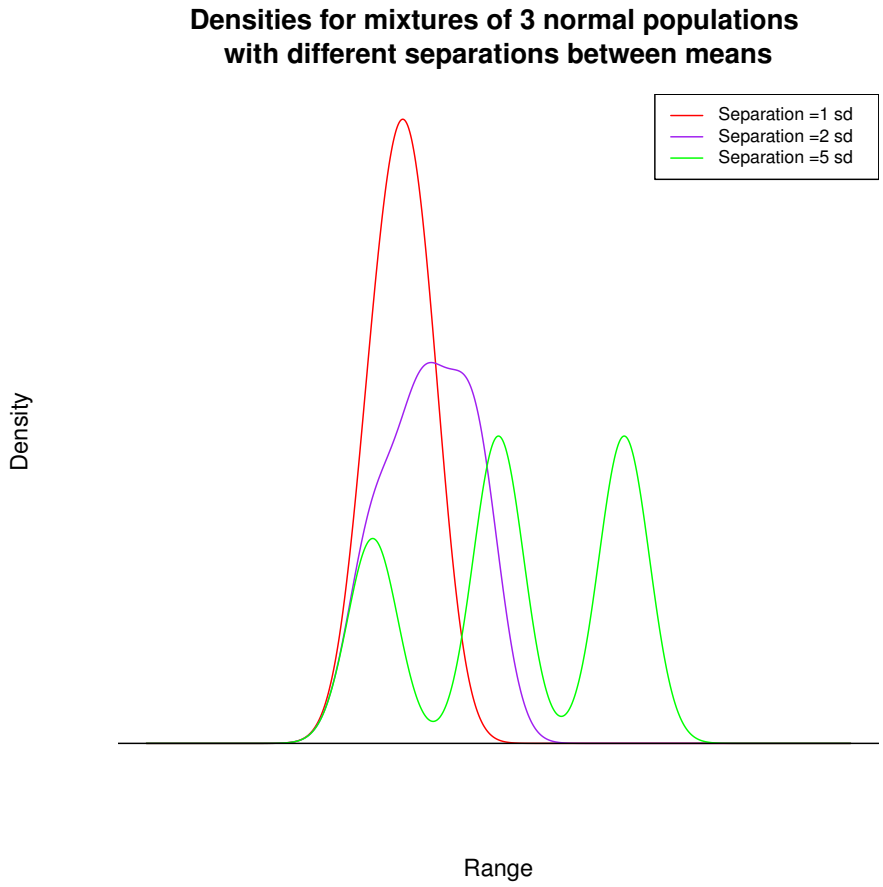

Figure 4-3.: Plot of the densities of a mixture of three normal distributions, with weight of 25 %, 37.5 % and 37.5 %. Source: build by the authors

Hellinger Distance for a mixture of three normal populations, Part 1 of 2.

	Parameters				Mean		SD	
#	Sample Size	Separation	Weight		GA	EM	GA	EM
1	30	1	5	2x47,5	NA	19,3487	NA	13,8950
2	50	1	5	2x47,5	NA	14,5957	NA	11,0491
3	100	1	5	2x47,5	129,5250	11,8703	3,9881	10,1206
4	200	1	5	2x47,5	128,2186	9,3690	1,7679	8,0910
5	30	2	5	2x47,5	NA	16,5984	NA	12,5496
6	50	2	5	2x47,5	NA	13,8891	NA	11,1954
7	100	2	5	2x47,5	109,3405	12,4531	10,2299	11,1815
8	200	2	5	2x47,5	109,4798	11,2289	3,7445	10,2198
9	30	5	5	2x47,5	NA	12,1189	NA	11,9492
10	50	5	5	2x47,5	98,2814	11,1436	31,8425	14,1184
11	100	5	5	2x47,5	79,8094	8,1169	15,0396	13,4625
12	200	5	5	2x47,5	87,3315	6,9644	12,9044	13,0165
13	30	1	10	2x45	NA	19,3208	NA	13,7580
14	50	1	10	2x45	126,1618	15,0075	5,3199	12,3377
15	100	1	10	2x45	125,0823	12,2665	3,6767	11,0550
16	200	1	10	2x45	123,9831	10,6264	1,9318	10,3144
17	30	2	10	2x45	NA	16,8123	NA	13,0197
18	50	2	10	2x45	106,8154	13,7236	14,0258	11,0697
19	100	2	10	2x45	102,6485	12,1146	8,3005	12,0258
20	200	2	10	2x45	102,2757	11,2259	3,0680	10,4774
21	30	5	10	2x45	78,9178	11,1852	30,0732	12,6397
22	50	5	10	2x45	77,8885	10,1085	17,9741	14,4147
23	100	5	10	2x45	78,8558	7,8781	13,6218	13,6624
24	200	5	10	2x45	80,9086	6,3508	11,6091	14,4297

Table 4-5.: Hellinger estimated distance for a mixture of three normal populations and the number of populations is known, part 1 . The second part is in Table**4-6** in page 30. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

Hellinger distance computed by Evolutive algorithms, GA, and traditional methods, EM it is shown in Tables **4-5** and **4-6** and graphic in Figure **4-4**. In this case it can be observed that the GA could not estimate the parameters in cases when the sample size was 30 and 50 items and the weight of one population was 5 % and 10 %. In this mixture, for all cases, as the separation increases, the mean HD decreases, except for the results of the GA when the weight is 50 % and sample sizes bigger than 50 items, because of the separation of 2 standard deviations (sd) was higher than the ones with 1 sd. Both the GA and the EM had better results when the composition of the mixture is even, the weight was 25 %. It can be concluded that the EM had better performance for the estimation of the parameters in a mixture of three normal populations with the settings evaluated in this scenario, because the means HD were smaller than the ones for the GA.

Hellinger Distance for a mixture of three normal populations, Part 2 of 2.								
	Parameters				Mean		SD	
#	Sample Size	Separation	Weight		GA	EM	GA	EM
25	30	1	25	2x37,5	115,0274	20,0112	6,9608	16,7180
26	50	1	25	2x37,5	113,0872	14,6814	4,8875	13,0309
27	100	1	25	2x37,5	111,9442	12,1571	4,3039	13,0794
28	200	1	25	2x37,5	110,9155	9,3455	1,8807	10,6120
29	30	2	25	2x37,5	98,6599	15,4575	15,3415	13,6241
30	50	2	25	2x37,5	96,2138	13,2369	15,4938	12,9151
31	100	2	25	2x37,5	92,4742	10,3372	11,2169	10,3576
32	200	2	25	2x37,5	90,6956	8,3961	3,9839	9,9917
33	30	5	25	2x37,5	74,5942	10,3042	29,3455	13,6233
34	50	5	25	2x37,5	69,4539	7,5522	25,5945	12,0071
35	100	5	25	2x37,5	62,7011	5,6851	9,5670	12,1097
36	200	5	25	2x37,5	73,2189	3,4947	5,3343	9,9318
37	30	1	50	2x25	103,6391	21,3477	10,7505	18,8318
38	50	1	50	2x25	102,9931	16,6304	10,4410	18,4549
39	100	1	50	2x25	102,0090	11,4809	8,3348	13,7736
40	200	1	50	2x25	103,4303	10,0692	7,9564	13,0650
41	30	2	50	2x25	103,4473	16,1874	24,3480	14,3116
42	50	2	50	2x25	110,1336	12,3997	26,6109	11,7561
43	100	2	50	2x25	116,0325	8,8013	26,6599	9,2689
44	200	2	50	2x25	111,6243	6,9781	24,1235	8,1191
45	30	5	50	2x25	95,1602	10,0543	33,8949	9,5359
46	50	5	50	2x25	93,2880	7,9199	32,1312	9,2770
47	100	5	50	2x25	91,3741	5,3558	23,2679	8,4164
48	200	5	50	2x25	75,4049	4,9445	10,7698	9,2909

Table 4-6.: Hellinger estimated distance for a mixture of three normal populations and the number of populations is known, part 2 . The first part is in Table**4-5** in page 29.The NA is when the data could not be computed for the algorithm. Source: Build by the authors

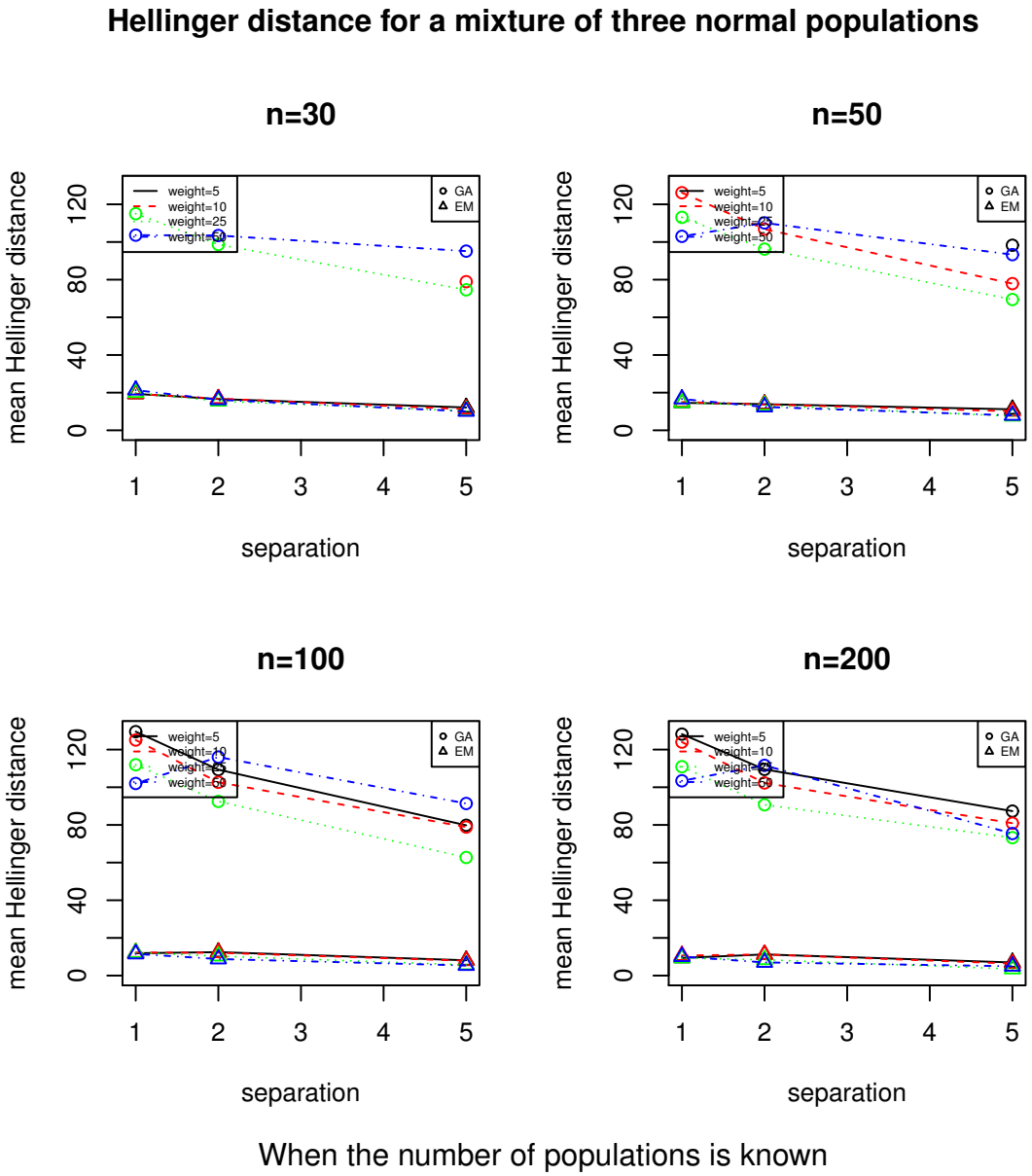

Figure 4-4.: Plot of the mean Hellinger distance for the mixture of 3 normal populations, in a mixture with known number of populations. Source: build by the authors

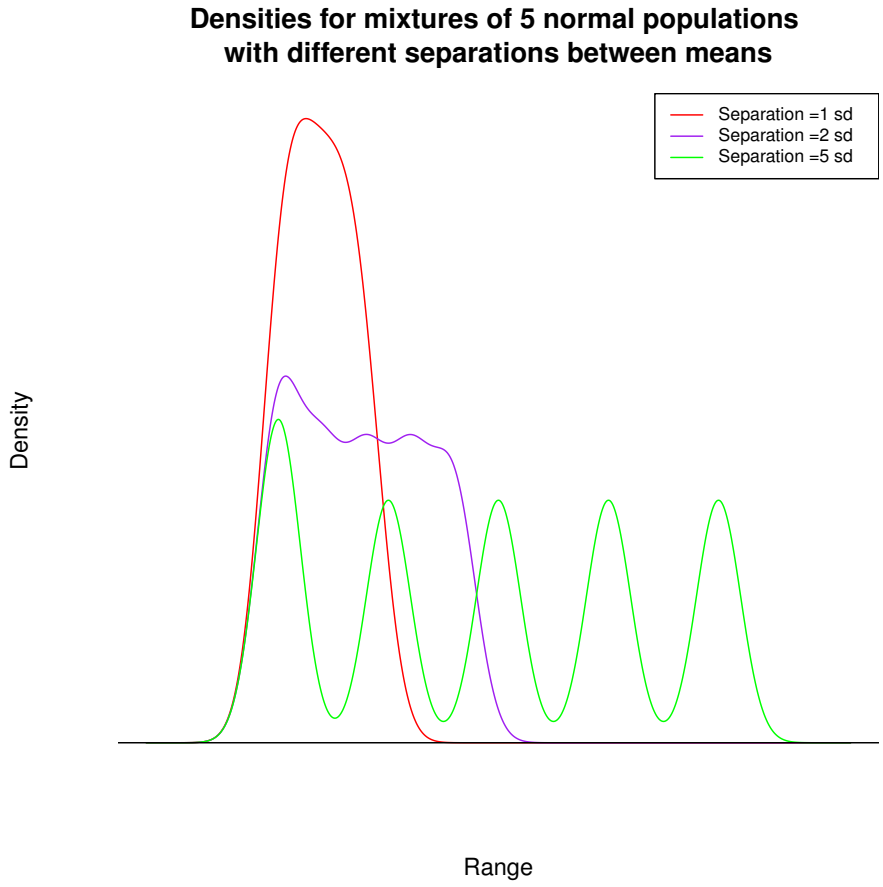

Figure 4-5.: Plot of the densities of a mixture of five normal distributions, with weight of 25 %, and four with weight 18.75 %. Source: build by the authors

4.1.3. Mixture of five normal distributions

For the mixture of five normal distributions created in this study, an example of the densities can be seen in Figure **4-5**, where the representations of a mixture with composition 25 %, and 4 populations with weight of 18.75 %, for different separations between means.

In the case of the mixture of 5 normal populations, for a known number of populations, the results of the simulations are summarized in Tables **4-7** and **4-8**, and the behavior of the data is shown in Figure **4-6**, on page 35.

The behavior of the mean Hellinger distance for the GA is influenced by separation because it increases as the separation increases, but this increment is less noticeable when the weights of the populations are even, arould 25 %. It is shown an increase on the mean HD when the weight is 50 % and the sample size increases, and that behaviour is not coherent with the ones shown in this case and the mixtures of two and three normal populations, this might suggest instability on the packages used. For the EM, the distance decreases with increases

Hellinger Distance for a mixture of five normal populations, Part 1 of 2.									
	Parameters					Mean		SD	
#	Sample Size	Separation	Weight			GA	EM	GA	EM
1	30	1	5	4x23,75		NA	14,8889	NA	6,2746
2	50	1	5	4x23,75		NA	7,7565	NA	3,7560
3	100	1	5	4x23,75		0,7973	3,7082	0,5804	1,7318
4	200	1	5	4x23,75		0,4127	1,6208	0,2657	0,7953
5	30	2	5	4x23,75		NA	11,2572	NA	5,5191
6	50	2	5	4x23,75		NA	5,1774	NA	2,3832
7	100	2	5	4x23,75		1,0847	2,5088	0,6323	1,1968
8	200	2	5	4x23,75		0,6228	1,2668	0,2862	0,6503
9	30	5	5	4x23,75		NA	7,2303	NA	2,6747
10	50	5	5	4x23,75		4,8132	4,4596	1,6782	1,5605
11	100	5	5	4x23,75		4,8643	2,2613	0,8004	1,0537
12	200	5	5	4x23,75		5,0323	1,5594	0,5212	0,9678
13	30	1	10	4x22,5		NA	14,9439	NA	6,7096
14	50	1	10	4x22,5		1,6824	8,3942	1,2918	4,0174
15	100	1	10	4x22,5		0,7809	3,5277	0,5484	1,5083
16	200	1	10	4x22,5		0,4305	1,5521	0,2951	0,7757
17	30	2	10	4x22,5		11,2170	10,8377	NA	4,5112
18	50	2	10	4x22,5		1,9771	5,4982	1,1243	2,5195
19	100	2	10	4x22,5		1,0637	2,5711	0,5772	1,1840
20	200	2	10	4x22,5		0,6269	1,1817	0,2969	0,5330
21	30	5	10	4x22,5		5,2709	7,0619	2,8471	2,5876
22	50	5	10	4x22,5		4,2011	4,4719	1,8182	1,6689
23	100	5	10	4x22,5		4,9406	2,5539	0,8620	1,2214
24	200	5	10	4x22,5		5,1724	1,7630	0,4937	1,1717

Table 4-7.: Hellinger estimated distance for a mixture of five normal populations and the number of populations is known, part 1 . The second part is in Table**4-8** in page 34. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

on the separation and sample size, the distances for the weights evaluated in this study are similar. As a conclusion, GA is a better alternative when the sample size and the separation between means are small, less than 50 items and 2 standard deviations, respectively, as long as all the populations in the mixture have more than 10 % weight. For the other experiments, the EM had better results.

Hellinger Distance for a mixture of five normal populations, Part 2 of 2.								
	Parameters				Mean		SD	
#	Sample Size	Separation	Weight		GA	EM	GA	EM
25	30	1	25	4x18,75	2,3508	14,5998	1,7067	6,4729
26	50	1	25	4x18,75	1,8241	8,1305	1,3000	3,4133
27	100	1	25	4x18,75	0,8601	3,4433	0,6078	1,4601
28	200	1	25	4x18,75	0,4554	1,7428	0,3047	1,5469
29	30	2	25	4x18,75	2,9130	10,1328	2,0184	4,6959
30	50	2	25	4x18,75	2,0954	5,8157	1,1683	2,6534
31	100	2	25	4x18,75	1,1763	2,5941	0,6417	1,2049
32	200	2	25	4x18,75	0,8931	1,1623	0,3768	0,5520
33	30	5	25	4x18,75	4,6563	7,0593	2,1528	2,5735
34	50	5	25	4x18,75	4,3249	4,4513	2,1370	1,5220
35	100	5	25	4x18,75	5,4291	2,4150	1,1700	1,1534
36	200	5	25	4x18,75	5,7990	1,3780	0,8236	0,9904
37	30	1	50	4x12,5	2,6853	14,8947	1,9672	6,9400
38	50	1	50	4x12,5	1,6963	8,0887	1,2692	3,9375
39	100	1	50	4x12,5	0,9057	3,6920	0,7012	1,7673
40	200	1	50	4x12,5	0,4756	1,6134	0,3026	0,7741
41	30	2	50	4x12,5	3,5109	11,1888	3,4619	4,5879
42	50	2	50	4x12,5	3,2366	6,0556	2,9231	2,8091
43	100	2	50	4x12,5	5,1750	2,6444	2,8931	1,3471
44	200	2	50	4x12,5	10,4189	1,2792	3,4097	0,5633
45	30	5	50	4x12,5	6,9759	7,0365	2,9229	2,2482
46	50	5	50	4x12,5	8,1814	4,7289	3,1531	1,5149
47	100	5	50	4x12,5	11,0188	2,8991	2,7923	0,8014
48	200	5	50	4x12,5	14,3128	2,1362	2,2120	0,6039

Table 4-8.: Hellinger estimated distance for a mixture of five normal populations and the number of populations is known, part 2 . The first part is in Table**4-7** in page 33. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

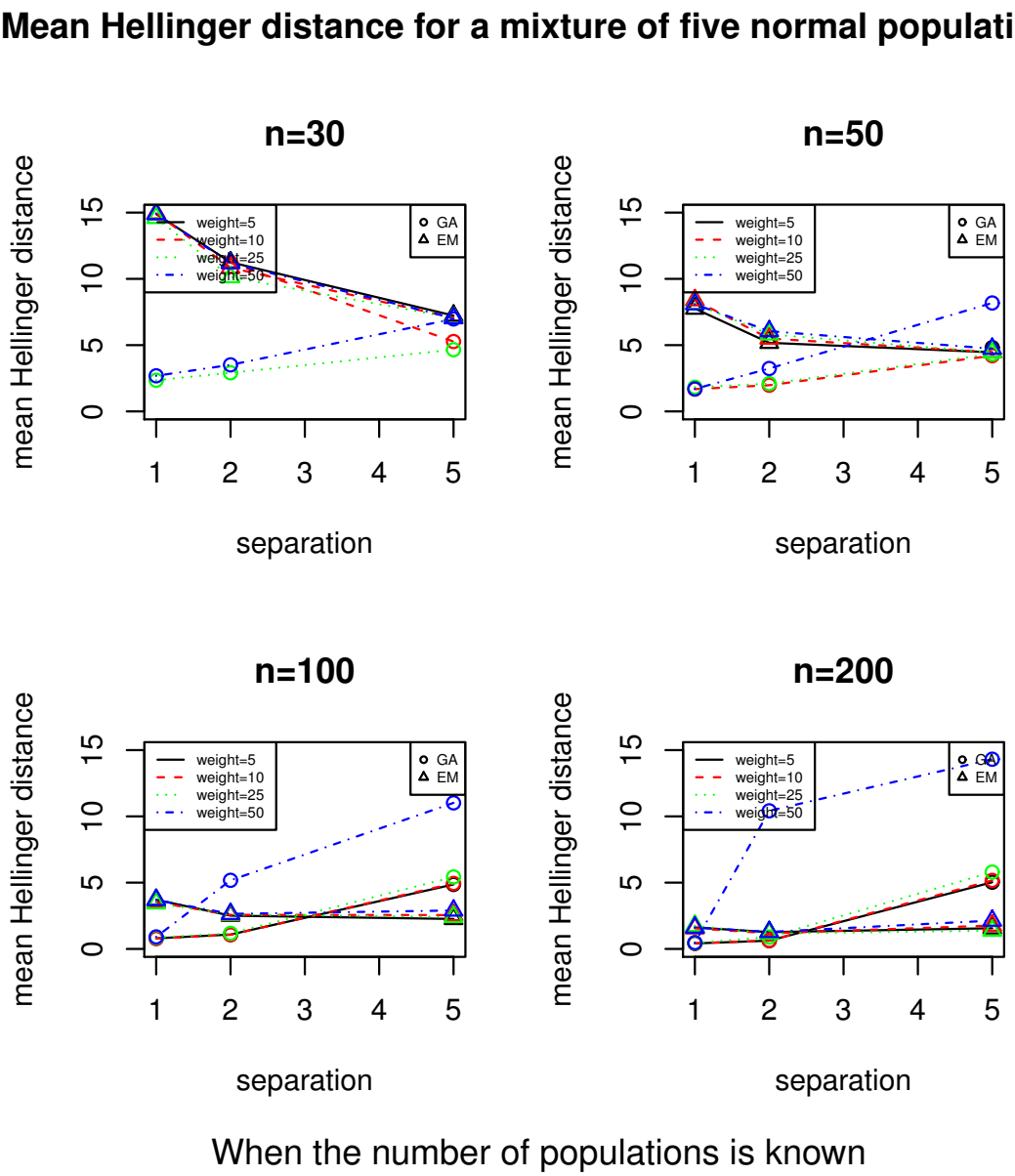

Figure 4-6.: Plot of the mean Hellinger distance for the mixture of 5 normal populations, in a mixture with known number of populations. Source: build by the authors

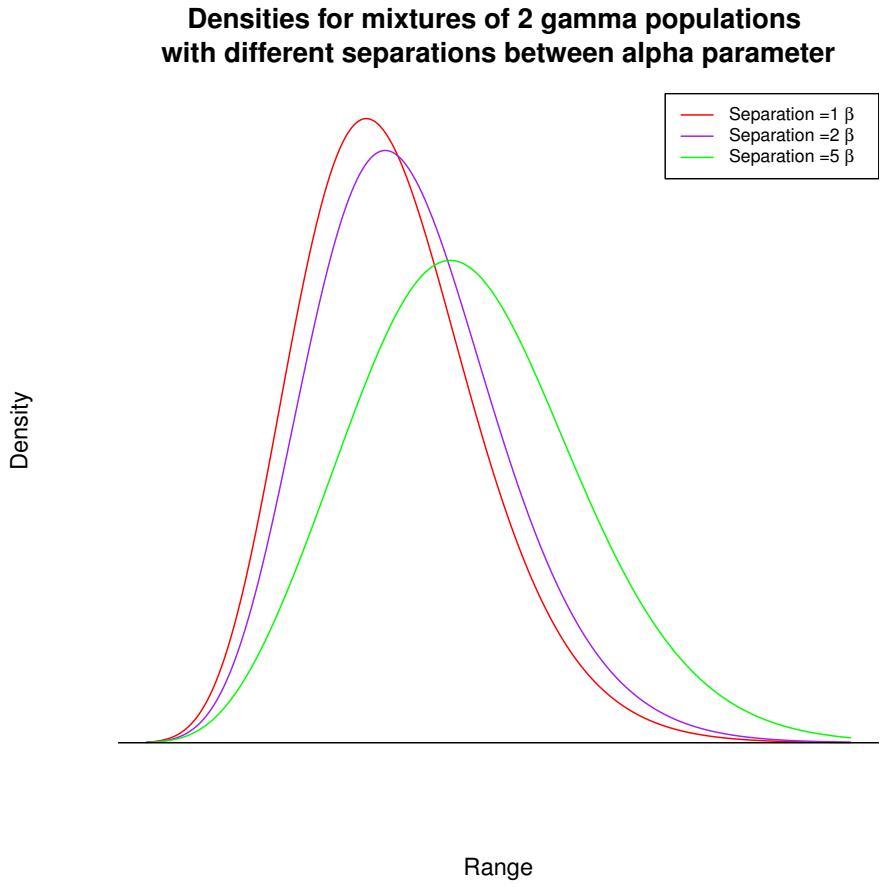

Figure 4-7.: Plot of the densities of a mixture of two gamma distributions, with weight of 25 %, and 75 %. Source: build by the authors

4.1.4. Mixture of two gamma distributions

For the mixture of two gamma distributions created in this study, an example of the densities can be seen in Figure **4-7**, where the representations of a mixture with composition 25 %, 75 %, for different separations between means.

The results for the estimation of the parameters of the mixture of gamma distributions it is shown in Tables **4-9**, page 37 and **4-10** on page 38, and their plots in Figure **4-8**. In the Tables and the figures it can be seen that for the gamma mixture of two populations, for the GA the separation between the α parameter increases the distance, also, the mean HD decreases when the weights in the mixture are even, or weight is 50 %, but for the EM the behaviour is the opposite, the separation decreases the mean HD. For the GA, the sample size does not have an obvious impact for the cases here studied, of 30, 50, 100 and 200 items, only the identification of all the populations in the mixture, because when the sample size was small and the weight was smaller than 10 %, the GA could not compute the parameters.

Hellinger Distance for a mixture of two gamma populations, Part 1 of 2.								
	Parameters				Mean		SD	
#	Sample Size	Separation	Weight		GA	EM	GA	EM
1	30	1	5	95	NA	68,5679	NA	34,8123
2	50	1	5	95	1,2692	57,4584	0,57458495	36,9736
3	100	1	5	95	0,1913	47,3056	0,1734	35,4738
4	200	1	5	95	0,0894	40,7156	0,0907	34,0546
5	30	2	5	95	NA	62,7838	NA	32,5497
6	50	2	5	95	1,7010	56,0702	0,6881	33,1209
7	100	2	5	95	0,2595	46,7824	0,2819	32,7800
8	200	2	5	95	0,1142	38,2714	0,1450	31,5179
9	30	5	5	95	NA	52,8592	NA	23,1625
10	50	5	5	95	5,9146	46,3868	1,0997	24,8910
11	100	5	5	95	0,4512	39,4411	0,4548	25,4519
12	200	5	5	95	0,2075	35,1988	0,2153	25,5347
13	30	1	10	90	3,1576	71,3746	1,8566	33,8742
14	50	1	10	90	0,4696	60,3630	0,4745	34,4689
15	100	1	10	90	0,1972	48,4631	0,2432	36,0426
16	200	1	10	90	0,0983	39,3559	0,1139	32,7873
17	30	2	10	90	4,0030	64,0057	3,3067	32,6848
18	50	2	10	90	0,6435	57,1330	0,6572	32,4226
19	100	2	10	90	0,2599	46,3885	0,2872	32,4753
20	200	2	10	90	0,1357	38,0838	0,1495	31,0215
21	30	5	10	90	NA	51,5740	NA	23,6459
22	50	5	10	90	1,0308	46,7276	1,2715	25,7810
23	100	5	10	90	0,5290	40,7789	0,5440	24,7764
24	200	5	10	90	0,3835	35,1254	0,3010	24,6896

Table 4-9.: Hellinger estimated distance for a mixture of two gamma populations and the number of populations is known, part 1. The second part is in Table**4-10** in page 38. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

For this mixture, the GA had better results for the cases when the weight was even, and the sample size was 200 because it could identify all the populations. Another thing important to notice was when the simulations were running, a convergence warning was shown for the EM algorithm very often, this could be related to the bigger distance reported in all the cases for the EM.

Hellinger Distance for a mixture of two gamma populations, Part 2 of 2.								
	Parameters				Mean		SD	
#	Sample Size	Separation	Weight		GA	EM	GA	EM
25	30	1	25	75	1,0517	71,6387	1,3507	34,1228
26	50	1	25	75	0,5865	62,3536	0,6761	36,3202
27	100	1	25	75	0,2909	53,3246	0,3125	35,9754
28	200	1	25	75	0,1585	39,4414	0,1707	32,5398
29	30	2	25	75	1,2838	68,7623	1,6026	29,9717
30	50	2	25	75	0,8298	61,4182	1,0021	32,8589
31	100	2	25	75	0,4983	51,7912	0,4995	33,2929
32	200	2	25	75	0,3405	41,8921	0,3063	32,1423
33	30	5	25	75	2,4328	52,0283	2,4950	24,4448
34	50	5	25	75	2,1809	49,8308	2,0270	24,3195
35	100	5	25	75	1,6818	43,9141	1,1405	24,9138
36	200	5	25	75	1,5131	38,9555	0,7746	24,8100
37	30	1	50	50	1,9559	76,6471	3,3184	32,2814
38	50	1	50	50	0,9508	68,8729	1,1200	34,4103
39	100	1	50	50	0,5059	60,0255	0,5201	36,6364
40	200	1	50	50	0,3069	47,0124	0,2804	36,0191
41	30	2	50	50	2,6116	70,5010	3,7573	30,4568
42	50	2	50	50	1,5198	67,4580	1,6601	30,6784
43	100	2	50	50	1,1224	58,8605	1,0807	33,9881
44	200	2	50	50	0,8178	45,1529	0,5651	32,5907
45	30	5	50	50	7,3237	57,0580	5,7427	21,8645
46	50	5	50	50	6,7262	56,2046	4,6662	22,0796
47	100	5	50	50	5,0898	46,2612	2,6142	25,1522
48	200	5	50	50	4,5522	39,3486	1,5601	24,9920

Table 4-10.: Hellinger estimated distance for a mixture of two gamma populations and the number of populations is known, part 2. The first part is in Table**4-9** in page 37. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

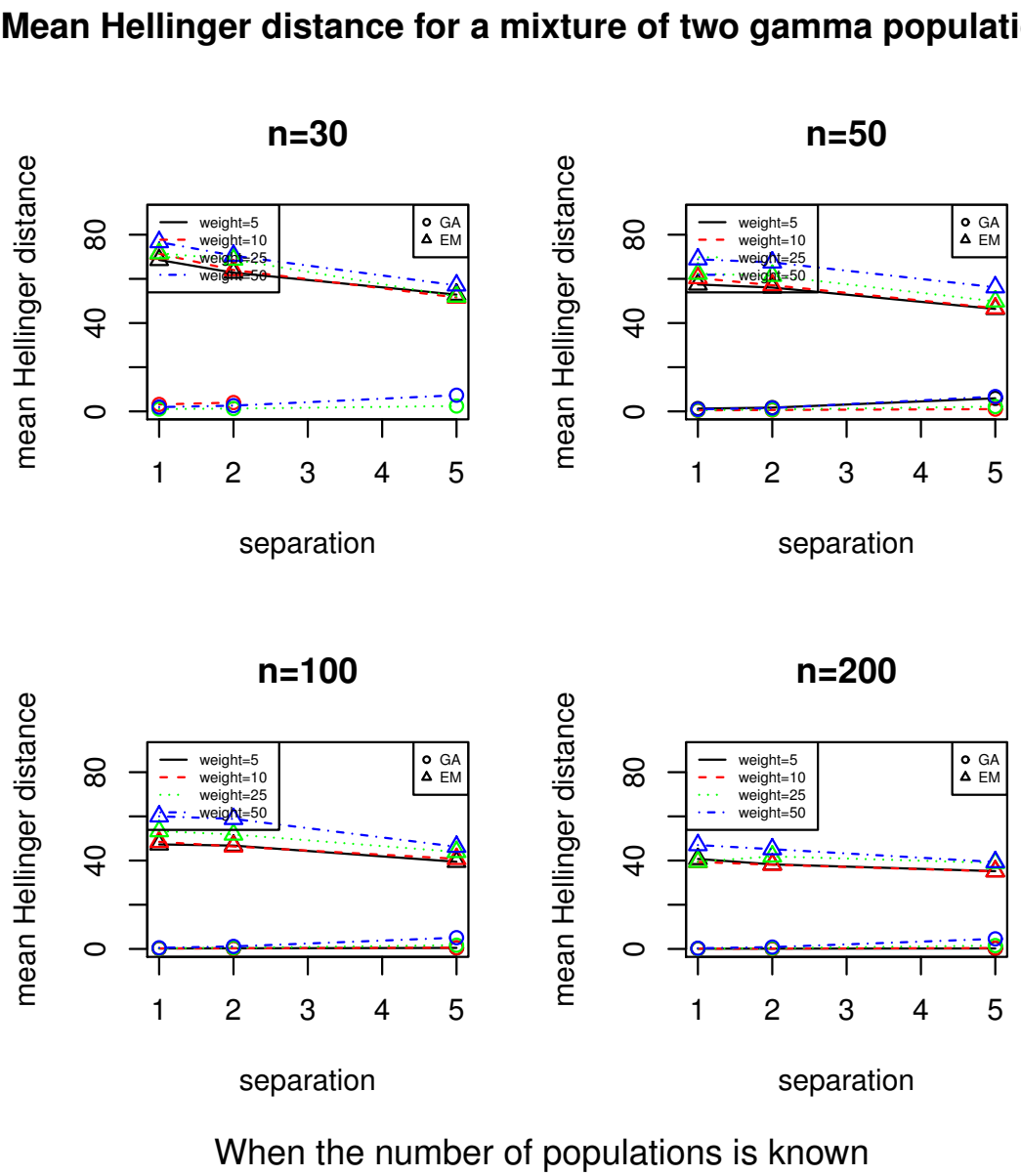

Figure 4-8.: Plot of the mean Hellinger distance for the mixture of 2 gamma populations, in a mixture with known number of populations. Source: build by the authors

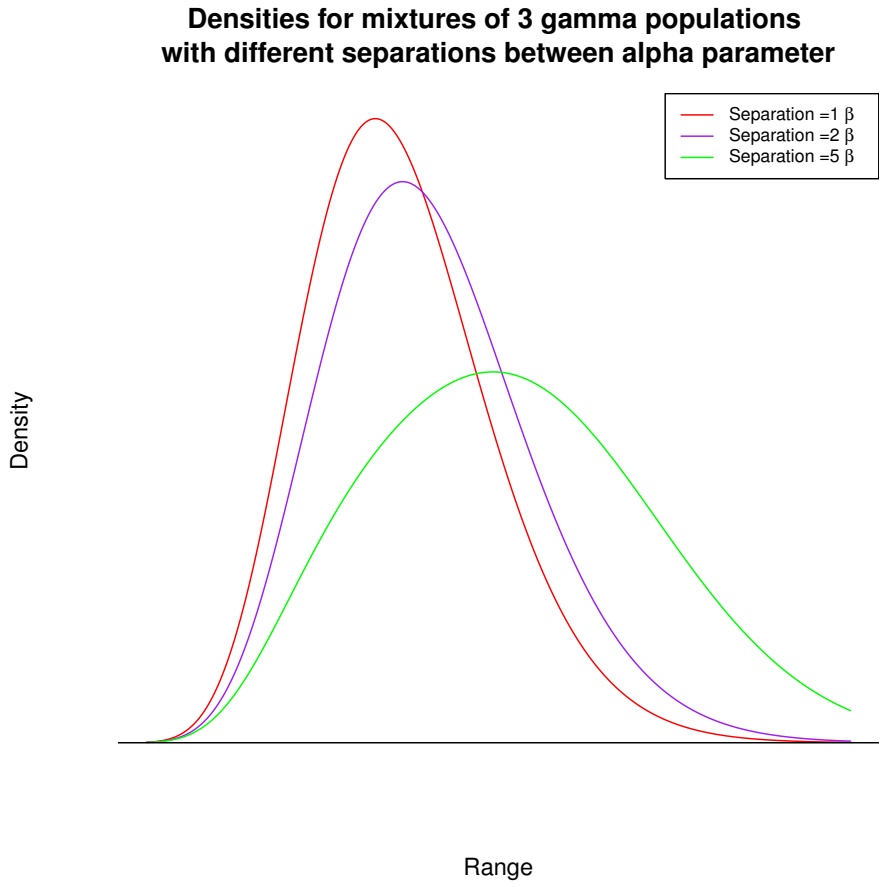

Figure 4-9.: Plot of the densities of a mixture of three gamma distributions, with weight of 25 %, and two with 38.5 %. Source: build by the authors

Mixture of three gamma distributions

For the mixture of three gamma distributions created in this study, an example of the densities can be seen in Figure **4-9**, where the representations of a mixture with composition 25 %, and two with 38.5 %, for different separations between means.

For the estimation of parameters of a mixture of three gamma distributions, the results are shown in Tables **4-11** and **4-12**, where the EM had multiple warning of convergence, more often than when the distribution of two gamma distributions is used. This can be seen in the Tables **4-11** and **4-12**, where the SD for the EM are bigger than the ones for the GA. The results shown in Figure **4-10** in page 43, it can be observed that the behavior is similar to the one in the mixture of two gammas, because of the GA could not detect in the cases when the sample size was smaller than 50 items and the weight of the at least one population was smaller than 10 %. The distance for the GA is also affected for the separation between the α parameters, because the mean HD increases when the separation increases, but decreases

Hellinger Distance for a mixture of three gamma populations, Part 1 of 2.									
	Parameters					Mean		SD	
#	Sample Size	Separation	Weight			GA	EM	GA	EM
1	30	1	5	2x47,5		NA	84,1558	NA	18,9398
2	50	1	5	2x47,5		13,2750	82,3475	9,7952649	21,8266
3	100	1	5	2x47,5		0,8238	80,5042	0,8456	23,6530
4	200	1	5	2x47,5		0,4421	76,6735	0,4057	26,3589
5	30	2	5	2x47,5		NA	73,7767	NA	15,9247
6	50	2	5	2x47,5		23,3999	73,0741	NA	17,2230
7	100	2	5	2x47,5		1,9031	68,8829	1,5393	22,1102
8	200	2	5	2x47,5		1,3498	67,7641	0,8285	21,9799
9	30	5	5	2x47,5		NA	50,2050	NA	9,1568
10	50	5	5	2x47,5		NA	49,4588	NA	10,7803
11	100	5	5	2x47,5		8,3971	48,7896	3,6137	11,6383
12	200	5	5	2x47,5		5,2914	48,2923	1,5947	12,1928
13	30	1	10	2x45		NA	85,0740	NA	18,0693
14	50	1	10	2x45		2,3120	81,8196	3,6897	21,8024
15	100	1	10	2x45		0,7315	77,8697	0,8659	25,6486
16	200	1	10	2x45		0,5315	74,1404	0,4730	28,3004
17	30	2	10	2x45		NA	74,3094	NA	14,4471
18	50	2	10	2x45		3,8816	71,9952	4,1422	17,8308
19	100	2	10	2x45		1,8869	71,0125	1,5370	19,2306
20	200	2	10	2x45		1,6387	65,8931	0,9856	23,6551
21	30	5	10	2x45		NA	50,0846	NA	8,6084
22	50	5	10	2x45		10,9170	49,0181	6,7602	10,5745
23	100	5	10	2x45		7,0357	49,4859	2,7668	9,9866
24	200	5	10	2x45		6,2828	47,7187	1,8558	12,9637

Table 4-11.: Hellinger estimated distance for a mixture of three gamma populations and the number of populations is known, part 1. The second part is in Table**4-12** in page 38. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

when the composition of the mixture, the weights, are even. The sample size does not have an evident impact in the HD. For the EM, the increase in the separation and sample size improved the performance of the algorithm, decreasing the distance. In general, EM had better results detecting all the populations, but GA had smaller distance when the sample size was big and the weight of each population in the mixture was bigger than 10 %.

Hellinger Distance for a mixture of three gamma populations, Part 2 of 2.								
	Parameters				Mean		SD	
#	Sample Size	Separation	Weight		GA	EM	GA	EM
25	30	1	25	2x37,5	1,6448	83,0304	1,9484	20,5290
26	50	1	25	2x37,5	2,1718	81,7691	5,4029	22,4656
27	100	1	25	2x37,5	1,1177	80,1751	1,2438	23,5901
28	200	1	25	2x37,5	0,7352	75,6153	0,6910	27,0298
29	30	2	25	2x37,5	2,8297	74,7833	2,4589	13,5058
30	50	2	25	2x37,5	4,3019	71,3831	5,6036	18,3040
31	100	2	25	2x37,5	2,9742	69,9791	2,1311	18,6188
32	200	2	25	2x37,5	2,6464	68,2123	1,5202	21,4852
33	30	5	25	2x37,5	5,5937	49,6378	2,9235	9,7195
34	50	5	25	2x37,5	10,4963	49,5782	4,5611	9,2634
35	100	5	25	2x37,5	10,6615	48,5288	3,4881	10,9136
36	200	5	25	2x37,5	10,2250	47,1806	2,4801	12,5170
37	30	1	50	2x25	3,7111	85,7999	5,5999	15,8461
38	50	1	50	2x25	4,0349	84,3997	5,6781	16,9584
39	100	1	50	2x25	1,9286	80,2984	2,1270	22,9093
40	200	1	50	2x25	1,4076	79,6296	1,0814	23,6994
41	30	2	50	2x25	6,5694	73,7005	6,2974	14,8599
42	50	2	50	2x25	7,3119	72,3057	6,1670	16,3956
43	100	2	50	2x25	5,8583	72,4179	4,2803	15,5594
44	200	2	50	2x25	4,8452	68,7495	2,6273	20,2917
45	30	5	50	2x25	17,2319	49,2159	6,8145	8,3722
46	50	5	50	2x25	18,5889	49,3768	6,7351	7,6509
47	100	5	50	2x25	18,3145	48,1269	5,6769	10,3046
48	200	5	50	2x25	17,9229	47,7042	3,8315	10,2017

Table 4-12.: Hellinger estimated distance for a mixture of three gamma populations and the number of populations is known, part 2. The first part is in Table**4-11** in page 41. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

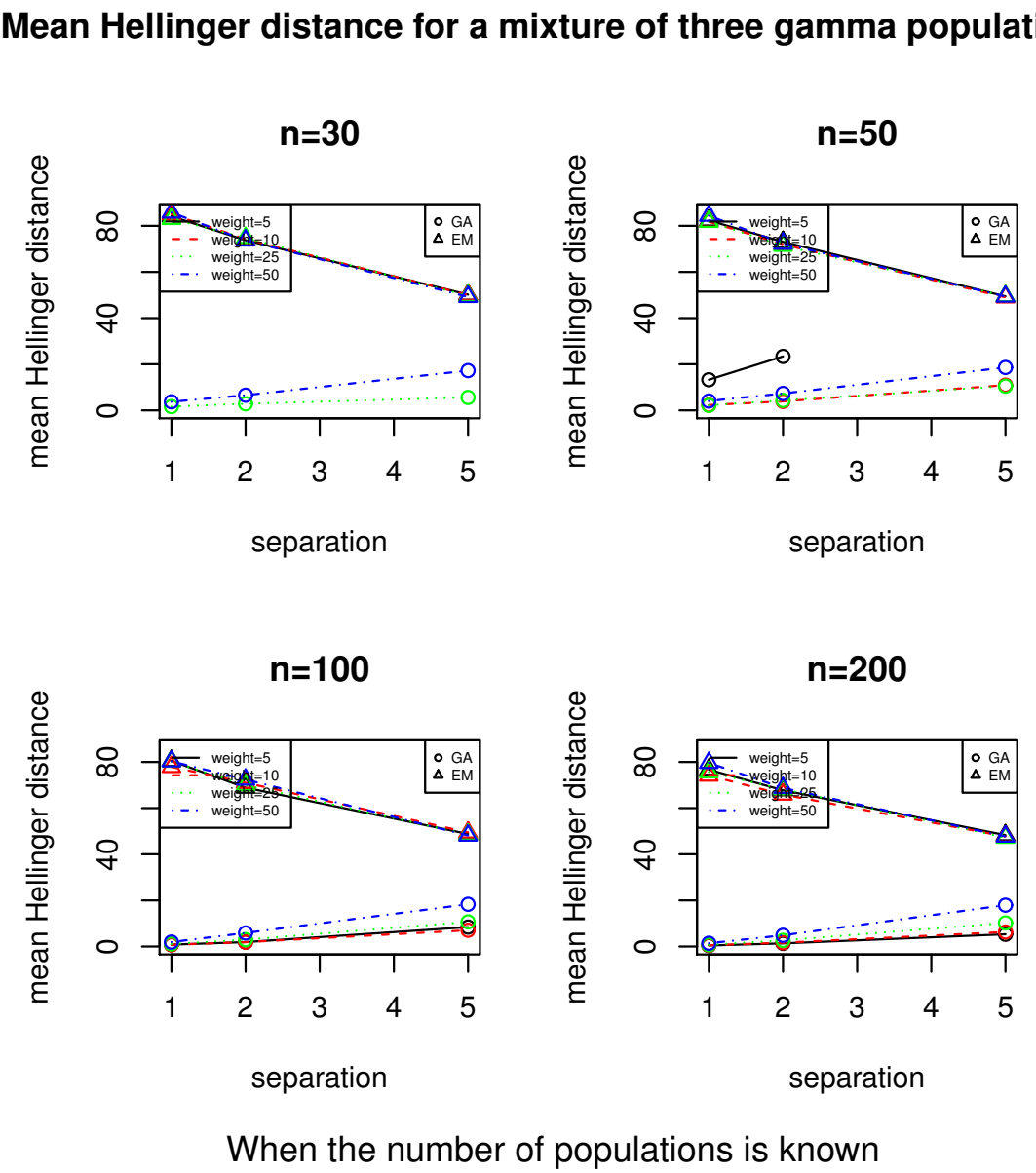

Figure 4-10.: Plot of the mean Hellinger distance for the mixture of 3 gamma populations, in a mixture with known number of populations. Source: build by the authors

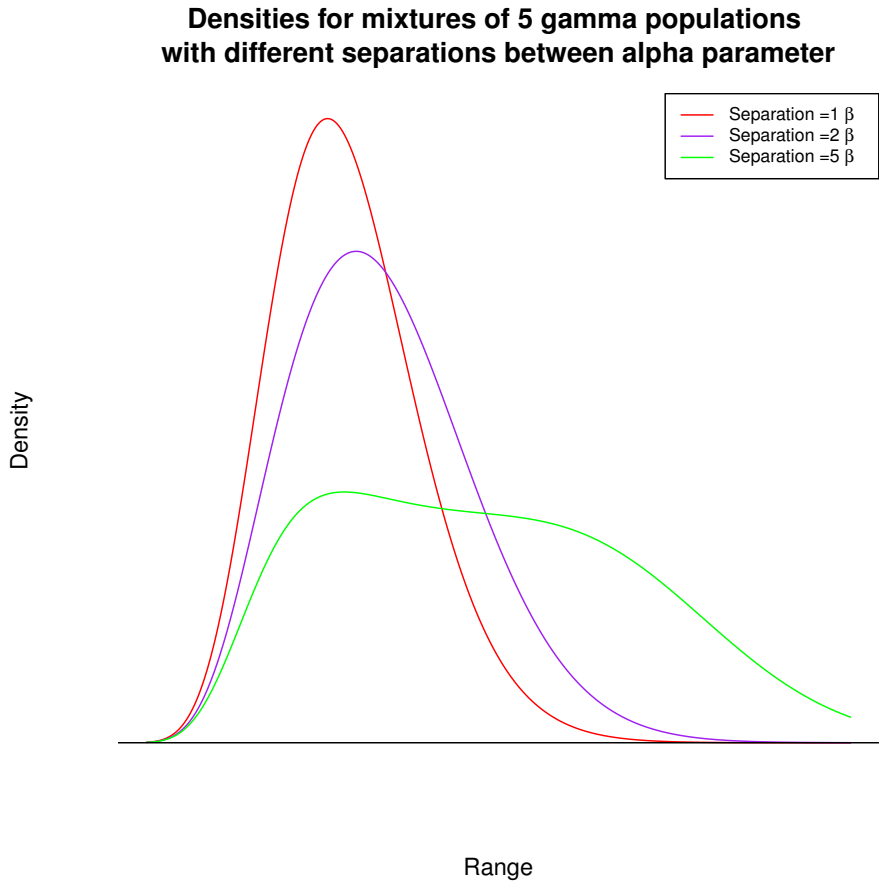

Figure 4-11.: Plot of the densities of a mixture of five gamma distributions, with weight of 25 %, and four with 18.75 %. Source: build by the authors

Mixture of five gamma distributions

For the mixture of five gamma distributions created in this study, an example of the densities can be seen in Figure **4-11**, where the representations of a mixture with composition 25 %, and four with 18.75 %, for different separations between means.

The results of the simulation for the estimation of parameters of a mixture of five gamma distributions are displayed in Tables **4-13** and **4-14**, and their plot in Figure **4-12** in page 47. The EM algorithm presented a lack of convergence, given by warnings with the results, as mentioned in previous sections, but in general, the separation between the populations in the mixture decreases the estimated distance. On the other hand, the GA method yielded NA as a result of some of the experiments, it is because it could not detect one or more populations, and this happened when there was a small sample size, 50 items or less, small weight, 10 % or less and sample size, it was needed more than 100 items to successfully estimate the parameters. When the separation is small, the difference between the GA and

Hellinger Distance for a mixture of five gamma populations, Part 1 of 2.									
	Parameters					Mean		SD	
#	Sample Size	Separation	Weight			GA	EM	GA	EM
1	30	1	5	4x23,75		NA	77,8143	NA	8,1779
2	50	1	5	4x23,75		NA	77,0358	NA	10,9278
3	100	1	5	4x23,75		4,9134	78,9741	4,3588	6,6175
4	200	1	5	4x23,75		2,6192	79,3615	2,0957	6,6740
5	30	2	5	4x23,75		NA	59,5560	NA	8,4743
6	50	2	5	4x23,75		NA	60,9448	NA	4,4329
7	100	2	5	4x23,75		12,7936	61,4185	6,1951	3,1243
8	200	2	5	4x23,75		7,2520	61,1797	2,8717	4,8548
9	30	5	5	4x23,75		NA	34,3061	NA	1,8943
10	50	5	5	4x23,75		NA	34,1856	NA	2,6224
11	100	5	5	4x23,75		21,2955	34,2539	1,0479	2,7060
12	200	5	5	4x23,75		13,4434	34,5939	2,5884	1,4037
13	30	1	10	4x22,5		80,0212	79,5075	NA	3,0217
14	50	1	10	4x22,5		1,8289	79,0298	1,4048	5,6764
15	100	1	10	4x22,5		3,6608	78,8449	3,5228	6,8807
16	200	1	10	4x22,5		3,0102	79,5152	2,1022	4,0307
17	30	2	10	4x22,5		NA	59,7416	NA	8,0443
18	50	2	10	4x22,5		3,5338	60,0580	2,1517	6,0446
19	100	2	10	4x22,5		8,9449	60,9756	5,0401	4,1156
20	200	2	10	4x22,5		8,6232	60,7908	3,4083	5,2973
21	30	5	10	4x22,5		NA	34,0302	NA	3,1266
22	50	5	10	4x22,5		NA	33,7865	NA	3,6960
23	100	5	10	4x22,5		12,8008	34,1561	2,9656	2,7073
24	200	5	10	4x22,5		14,9101	34,4405	2,3107	1,7520

Table 4-13.: Hellinger estimated distance for a mixture of five gamma populations and the number of populations is known, part 1. The second part is in Table**4-14** in page 46.The NA is when the data could not be computed for the algorithm. Source: Build by the authors

the EM is more noticeable, but in all cases the GA had better results than the EM. For this reason, the main conclusion is that the GA is a better option than the EM to estimate the parameters in these mixtures of gamma populations, when the sample size is 100 items or bigger.

Hellinger Distance for a mixture of five gamma populations, Part 2 of 2.									
	Parameters					Mean		SD	
#	Sample Size	Separation		Weight		GA	EM	GA	EM
25	30	1	25	4x18,75		4,4590	76,7364	4,8283	8,7132
26	50	1	25	4x18,75		4,1631	77,3495	4,2504	9,2041
27	100	1	25	4x18,75		5,7485	78,1895	5,2837	6,6486
28	200	1	25	4x18,75		4,4806	78,6670	2,9821	4,9395
29	30	2	25	4x18,75		6,3456	57,4868	4,3838	10,8427
30	50	2	25	4x18,75		8,4485	59,8398	4,8973	4,7367
31	100	2	25	4x18,75		12,3370	60,0828	5,6930	4,2870
32	200	2	25	4x18,75		11,7142	60,0021	4,1816	5,0271
33	30	5	25	4x18,75		10,6893	33,7938	1,9663	0,7638
34	50	5	25	4x18,75		11,6609	33,4628	2,3120	2,7914
35	100	5	25	4x18,75		17,1391	33,4802	2,8017	2,7948
36	200	5	25	4x18,75		17,6964	33,8689	2,1216	0,9734
37	30	1	50	4x12,5		5,2594	77,9754	7,4966	6,8389
38	50	1	50	4x12,5		3,9818	77,0486	3,7523	9,9709
39	100	1	50	4x12,5		5,3634	77,7322	5,3193	8,5890
40	200	1	50	4x12,5		4,0107	78,5684	2,8076	4,8992
41	30	2	50	4x12,5		NA	53,8749	NA	11,0343
42	50	2	50	4x12,5		15,7958	57,9101	7,8302	7,6553
43	100	2	50	4x12,5		17,9096	58,0411	6,9288	7,2743
44	200	2	50	4x12,5		18,5490	58,6724	6,1486	5,6559
45	30	5	50	4x12,5		NA	33,0168	NA	0,0181
46	50	5	50	4x12,5		20,4938	32,9384	3,4448	0,5390
47	100	5	50	4x12,5		24,0177	32,7804	2,9680	1,6480
48	200	5	50	4x12,5		23,8127	32,8311	2,2869	1,3841

Table 4-14.: Hellinger estimated distance for a mixture of five gamma populations and the number of populations is known, part 2. The first part is in Table**4-13** in page 45. The NA is when the data could not be computed for the algorithm. Source: Build by the authors

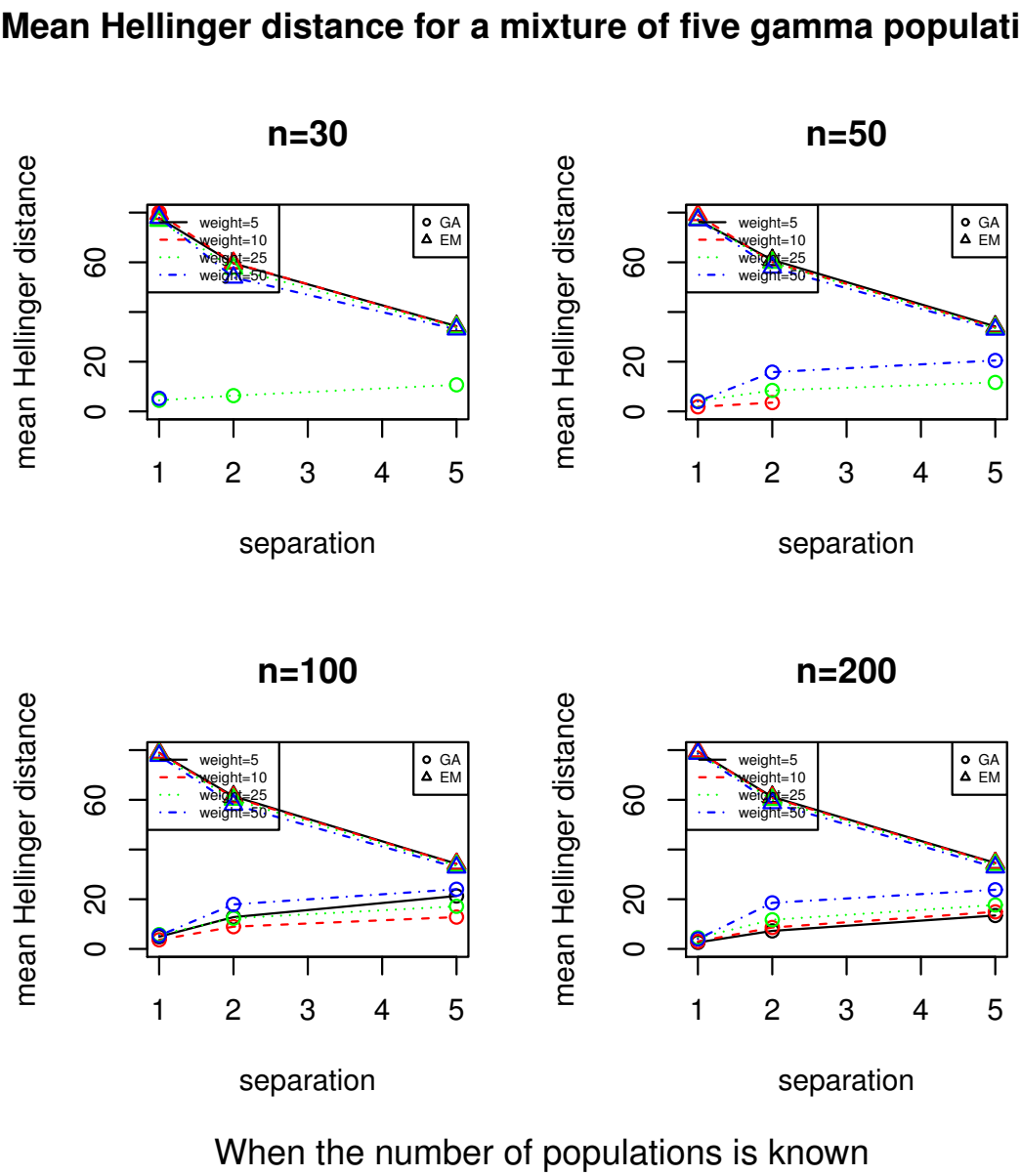

Figure 4-12.: Plot of the mean Hellinger distance for the mixture of 5 gamma populations, in a mixture with known number of populations. Source: build by the authors

4.2. Number of populations unknown

4.2.1. Mixture of two normal populations

For this scenario the data were simulated as a mixture of normal populations with the parameters shown in Tables **4-15** and **4-16**. The mean number of populations estimated for the GA and the EM algorithm and their standard deviation is shown in the same tables, and the behavior of the estimation in the Figure **4-13** in page 51. The initial number of populations was set at 5 for both methods. In the Tables and the Figures it can be observed that both methods diverge in their behavior, because the GA overestimates and the EM underestimates the number of populations. For the GA the closest experiments were the ones with small sample data, this could be for the same reason when the number of populations was known, when they could not estimate all the data and yielded NaN, as it is shown in Section 4.1.1 in page 24.When the sample size was big enough, 100 or 200 the method in all the cases estimated 5 populations, because the standard deviation was 0. For the EM algorithm, all the results were close, but the best results were obtained when the identification was easy, as exposed in the previous analysis, when the sample size, separation and weight was large. In the other cases, the method could not detect all the populations, also, this method had a smaller variation compared with the GA. Because of that reason, the conclusion is that the GA is worse than the EM in the estimation of the number of populations in a mixture of normal distributions.

Two known populations - Part 1/2				*Number of populations for Normal Distribution*				
	Parameters			**Mean**		**SD**		
#	**Sample Size**	**separation**	**weight**	**GA**	**EM**	**GA**	**EM**	
1	30	1	5	3,594	1,004	0,691		0,063
2	50	1	5	4,404	1,000	0,605		0,000
3	100	1	5	4,980	1,000	0,140		0,000
4	200	1	5	5,000	1,000	0,000		0,000
5	30	2	5	3,476	1,004	0,726		0,063
6	50	2	5	4,252	1,000	0,633		0,000
7	100	2	5	4,982	1,000	0,133		0,000
8	200	2	5	5,000	1,000	0,000		0,000
9	30	5	5	3,142	1,054	0,671		0,289
10	50	5	5	3,832	1,020	0,690		0,178
11	100	5	5	4,956	1,004	0,205		0,063
12	200	5	5	5,000	1,000	0,000		0,000
13	30	1	10	3,630	1,006	0,697		0,077
14	50	1	10	4,340	1,002	0,617		0,045
15	100	1	10	4,990	1,000	0,100		0,000
16	200	1	10	5,000	1,000	0,000		0,000
17	30	2	10	3,460	1,008	0,736		0,089
18	50	2	10	4,198	1,000	0,654		0,000
19	100	2	10	4,992	1,000	0,089		0,000
20	200	2	10	5,000	1,000	0,000		0,000
21	30	5	10	2,872	1,230	0,719		0,534
22	50	5	10	3,982	1,180	0,750		0,460
23	100	5	10	4,968	1,084	0,176		0,305
24	200	5	10	5,000	1,048	0,000		0,232

Table 4-15.: Total populations calculated for a mixture of two normal distributions, part 1. The second part is in Table**4-16** in page 50. Source: Build by the authors

Two known populations - Part 2/2				*Number of populations for Normal Distribution*				
	Parameters			**Mean**		**SD**		
#	**Sample Size**	**separation**	**weight**	**GA**	**EM**	**GA**	**EM**	
25	30	1	25	3,574	1,006	0,719		0,077
26	50	1	25	4,312	1,000	0,660		0,000
27	100	1	25	4,982	1,000	0,133		0,000
28	200	1	25	5,000	1,000	0,000		0,000
29	30	2	25	3,346	1,010	0,769		0,100
30	50	2	25	4,240	1,002	0,647		0,045
31	100	2	25	4,980	1,000	0,140		0,000
32	200	2	25	5,000	1,000	0,000		0,000
33	30	5	25	2,970	1,756	0,725		0,495
34	50	5	25	4,088	1,896	0,643		0,462
35	100	5	25	4,966	1,942	0,181		0,258
36	200	5	25	5,000	2,002	0,000		0,118
37	30	1	50	3,526	1,008	0,726		0,089
38	50	1	50	4,312	1,002	0,632		0,045
39	100	1	50	4,990	1,000	0,100		0,000
40	200	1	50	5,000	1,000	0,000		0,000
41	30	2	50	3,402	1,014	0,717		0,118
42	50	2	50	4,238	1,004	0,631		0,063
43	100	2	50	4,986	1,000	0,118		0,000
44	200	2	50	5,000	1,000	0,000		0,000
45	30	5	50	2,946	1,972	0,740		0,227
46	50	5	50	4,028	2,000	0,669		0,090
47	100	5	50	4,944	2,000	0,230		0,000
48	200	5	50	5,000	2,000	0,000		0,000

Table 4-16.: Total populations calculated for a mixture of two normal distributions, part 2. The first part is in Table**4-15** in page 49. Source: Build by the authors

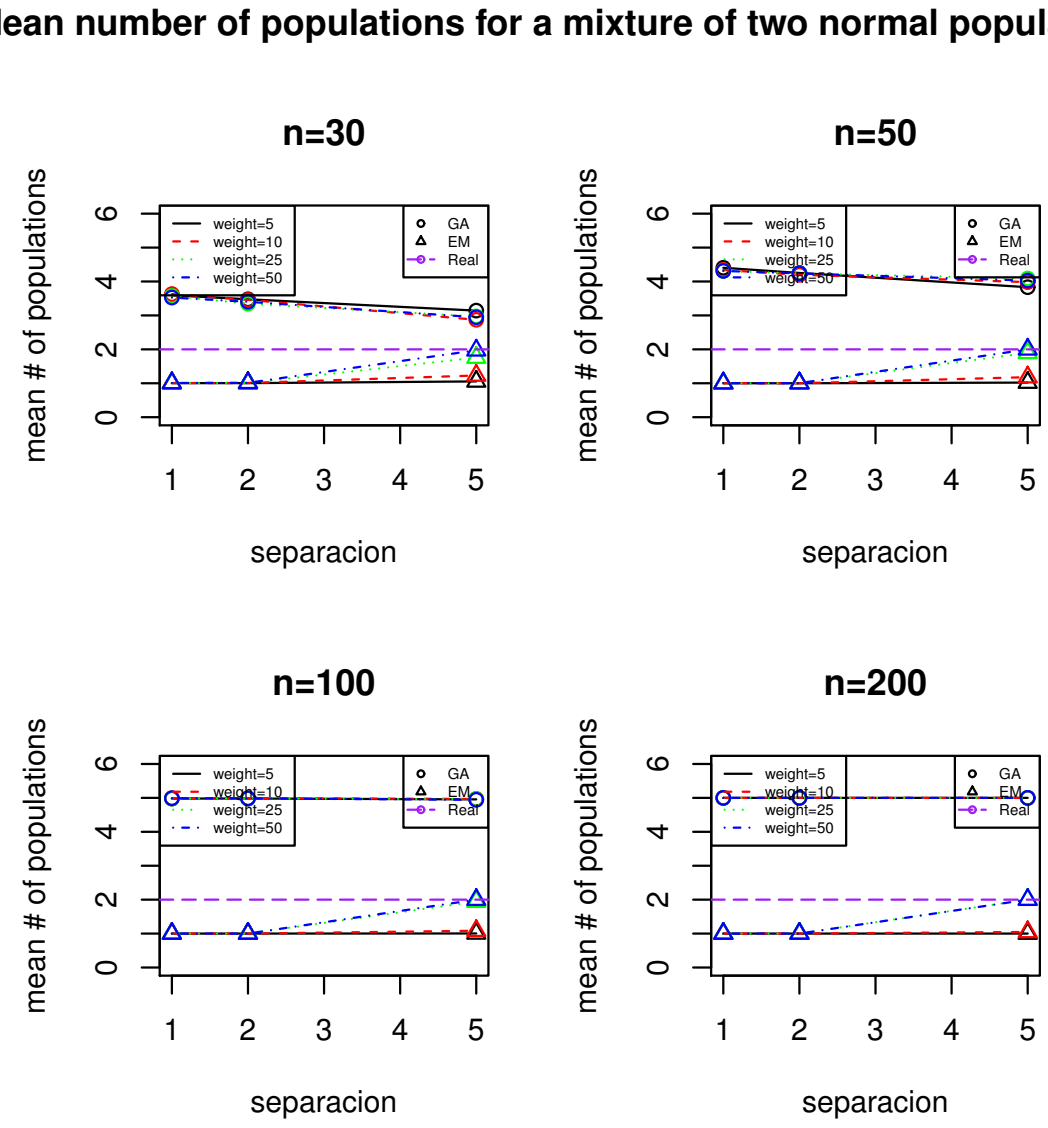

Figure 4-13.: Mean estimated number of populations for the mixture of two normal distributions. Source: Build by the authors

4.2.2. Mixture of three normal distributions

The results of the simulations for the estimation of the parameters of mixtures of three normal distributions can be observed in Tables **4-17** and **4-18**, and the Figure **4-14** in page 55. The starting value for the number of populations was 6 for both methods. As shown in the Tables and Figures, this case has the same behavior that the case with 2 populations. The GA overestimates and the EM underestimate the number of populations. The experiments with the closer result for the GA are those where there is small sample size, because it is unable to detect the populations with few data, but when there is plenty of data, such as 200, the algorithm always estimate the mixture with the initial parameter as the number of populations. The situation is different for the EM algorithm; because it underestimate the number of populations, the scenarios were it had the best performance were the ones with big sample and separation. It can be noticed that the EM algorithm had the smallest variation in all the scenarios, making it the most precise method among the studied ones, but it is not exact. Neither of methods in neither scenario had an exact performance, and this can be checked by looking carefuly the images.

Three known populations - Part 1/2				*Number of populations for Normal Distribution*				
		Parameters		**Mean**		**SD**		
	n	**separacion**	**weight**	**GA**	**EM**	**GA**	**EM**	
1	30	1	5	3,812	1,004	0,809		0,063
2	50	1	5	4,768	1,004	0,729		0,063
3	100	1	5	5,904	1,000	0,295		0,000
4	200	1	5	6,000	1,000	0,000		0,000
5	30	2	5	3,398	1,008	0,825		0,109
6	50	2	5	4,502	1,004	0,680		0,063
7	100	2	5	5,888	1,002	0,322		0,045
8	200	2	5	6,000	1,000	0,000		0,000
9	30	5	5	2,670	1,442	0,683		0,554
10	50	5	5	3,976	1,546	0,738		0,580
11	100	5	5	5,800	1,660	0,410		0,570
12	200	5	5	6,000	1,836	0,000		0,585
13	30	1	10	3,662	1,010	0,762		0,100
14	50	1	10	4,656	1,000	0,683		0,000
15	100	1	10	5,932	1,000	0,252		0,000
16	200	1	10	6,000	1,000	0,000		0,000
17	30	2	10	3,242	1,012	0,765		0,109
18	50	2	10	4,426	1,004	0,705		0,063
19	100	2	10	5,922	1,000	0,268		0,000
20	200	2	10	6,000	1,000	0,000		0,000
21	30	5	10	2,552	1,254	0,672		0,523
22	50	5	10	3,898	1,226	0,730		0,517
23	100	5	10	5,800	1,234	0,415		0,562
24	200	5	10	5,998	1,298	0,045		0,686

Table 4-17.: Total populations calculated for a mixture of three normal distributions, part 1. The second part is in Table**4-18** in page 54. Source: Build by the authors

Three known populations - Part 2/2				*Number of populations for Normal Distribution*				
		Parameters		**Mean**		**SD**		
	n	**separacion**	**weight**	**GA**	**EM**	**GA**	**EM**	
25	30	1	25	3,670	1,016	0,794		0,126
26	50	1	25	4,702	1,002	0,706		0,045
27	100	1	25	5,910	1,000	0,286		0,000
28	200	1	25	6,000	1,000	0,000		0,000
29	30	2	25	3,150	1,018	0,837		0,133
30	50	2	25	4,430	1,014	0,703		0,118
31	100	2	25	5,866	1,004	0,341		0,063
32	200	2	25	6,000	1,000	0,000		0,000
33	30	5	25	2,596	1,172	0,665		0,539
34	50	5	25	4,084	1,122	0,706		0,477
35	100	5	25	5,728	1,076	0,463		0,383
36	200	5	25	6,000	1,012	0,000		0,155
37	30	1	50	3,602	1,010	0,815		0,100
38	50	1	50	4,712	1,002	0,714		0,045
39	100	1	50	5,910	1,000	0,286		0,000
40	200	1	50	6,000	1,000	0,000		0,000
41	30	2	50	3,120	1,096	0,774		0,302
42	50	2	50	4,454	1,068	0,682		0,252
43	100	2	50	5,882	1,042	0,335		0,201
44	200	2	50	6,000	1,010	0,000		0,100
45	30	5	50	2,508	1,812	0,683		0,956
46	50	5	50	4,040	2,062	0,712		0,992
47	100	5	50	5,812	2,462	0,401		0,907
48	200	5	50	5,998	2,720	0,045		0,695

Table 4-18.: Total populations calculated for a mixture of three normal distributions, part 2. The first part is in Table**4-17** in page 53. Source: Build by the authors

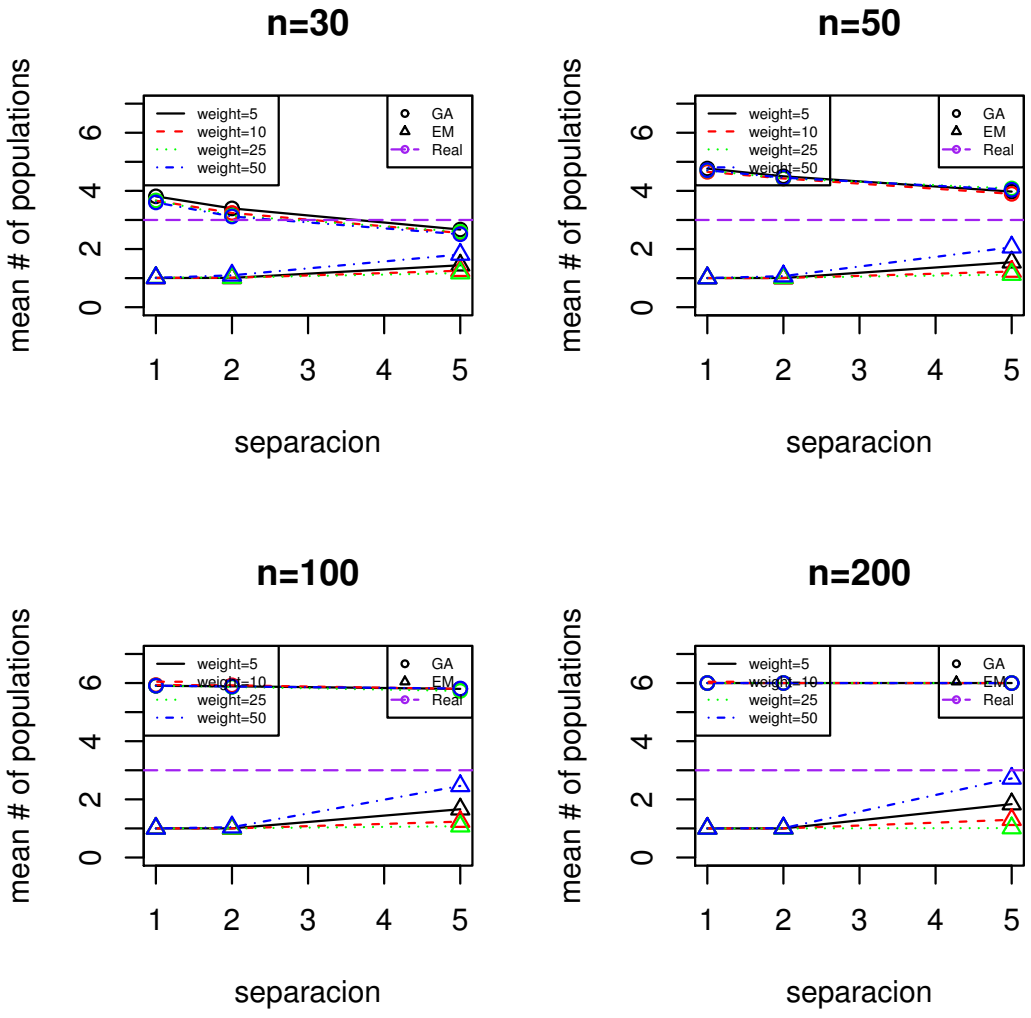

Figure 4-14.: Mean estimated number of populations for the mixture of three normal distributions. Source: Build by the authors

Five known populations - Part 1/2		*Number of populations for Normal Distribution*					
		Parameters		**Mean**		**SD**	
#	**n**	**separacion**	**weight**	**GA**	**EM**	**GA**	**EM**
1	30	1	5	2,984	1,010	0,909	0,100
2	50	1	5	5,072	1,002	0,830	0,045
3	100	1	5	7,212	1,000	0,660	0,000
4	200	1	5	7,998	1,000	0,045	0,000
5	30	2	5	2,072	1,062	0,767	0,250
6	50	2	5	4,376	1,032	0,802	0,176
7	100	2	5	6,800	1,024	0,667	0,153
8	200	2	5	7,988	1,002	0,109	0,045
9	30	5	5	1,554	1,184	0,616	0,432
10	50	5	5	3,740	1,150	0,763	0,374
11	100	5	5	6,448	1,160	0,587	0,367
12	200	5	5	7,994	1,174	0,077	0,379
13	30	1	10	2,924	1,024	0,867	0,153
14	50	1	10	5,040	1,002	0,865	0,045
15	100	1	10	7,196	1,000	0,680	0,000
16	200	1	10	7,996	1,000	0,063	0,000
17	30	2	10	1,986	1,040	0,729	0,196
18	50	2	10	4,250	1,030	0,765	0,171
19	100	2	10	6,774	1,010	0,660	0,100
20	200	2	10	7,990	1,004	0,100	0,063
21	30	5	10	1,548	1,120	0,604	0,325
22	50	5	10	3,630	1,120	0,674	0,337
23	100	5	10	6,400	1,074	0,556	0,262
24	200	5	10	7,982	1,058	0,133	0,234

Table 4-19.: Total populations calculated for a mixture of five normal distributions, part 1. The second part is in Table**4-20** in page 57. Source: Build by the authors

4.2.3. Mixture of five normal distributions

The result for the mean populations found for the mixture of 5 distributions can be found on Tables **4-19** and **4-20** and in Figure **4-15**, page 58. For this exercise the initial value for the number of populations was set to 8. It is shown in the figures that the number of populations was not close for neither of the methods.The GA tends to overestimate the number of populations when the size of the sample data allows it, but this time this algorithm underestimated the result more often when the sample size was small. Also, in general, the standard deviation was bigger than the one with 2 and 3 populations, respectively. For the EM algorithm in all the cases the results underestimate the real value, the only one close occurs when a population had the 50 % of the data and separation of 5 standard deviation, for the rest, it could not even detect that there was a mixture. As a conclusion, then neither of the methods could correctly estimate the number when there was a big amount of populations.

Five known populations - Part 2/2				*Number of populations for Normal Distribution*				
		Parameters		**Mean**		**SD**		
#	**n**	**separacion**	**weight**	**GA**	**EM**	**GA**	**EM**	
25	30	1	25	2,822	1,018	0,892		0,133
26	50	1	25	4,964	1,006	0,886		0,077
27	100	1	25	7,136	1,002	0,671		0,045
28	200	1	25	7,996	1,000	0,063		0,000
29	30	2	25	1,918	1,182	0,749		0,391
30	50	2	25	4,196	1,134	0,822		0,353
31	100	2	25	6,850	1,122	0,702		0,334
32	200	2	25	7,988	1,120	0,109		0,325
33	30	5	25	1,488	1,308	0,592		0,535
34	50	5	25	3,692	1,340	0,776		0,604
35	100	5	25	6,450	1,428	0,613		0,688
36	200	5	25	7,978	1,508	0,147		0,787
37	30	1	50	2,780	1,032	0,886		0,176
38	50	1	50	4,884	1,008	0,863		0,089
39	100	1	50	7,170	1,002	0,668		0,045
40	200	1	50	7,996	1,000	0,063		0,000
41	30	2	50	1,978	1,390	0,761		0,512
42	50	2	50	4,336	1,438	0,815		0,528
43	100	2	50	7,038	1,518	0,711		0,508
44	200	2	50	7,990	1,650	0,100		0,486
45	30	5	50	1,482	1,942	0,592		0,790
46	50	5	50	3,680	2,134	0,742		0,713
47	100	5	50	6,896	2,510	0,747		0,874
48	200	5	50	7,964	2,630	0,186		0,866

Table 4-20.: Total populations calculated for a mixture of five normal distributions, part 2. The first part is in Table**4-19** in page 56. Source: Build by the authors

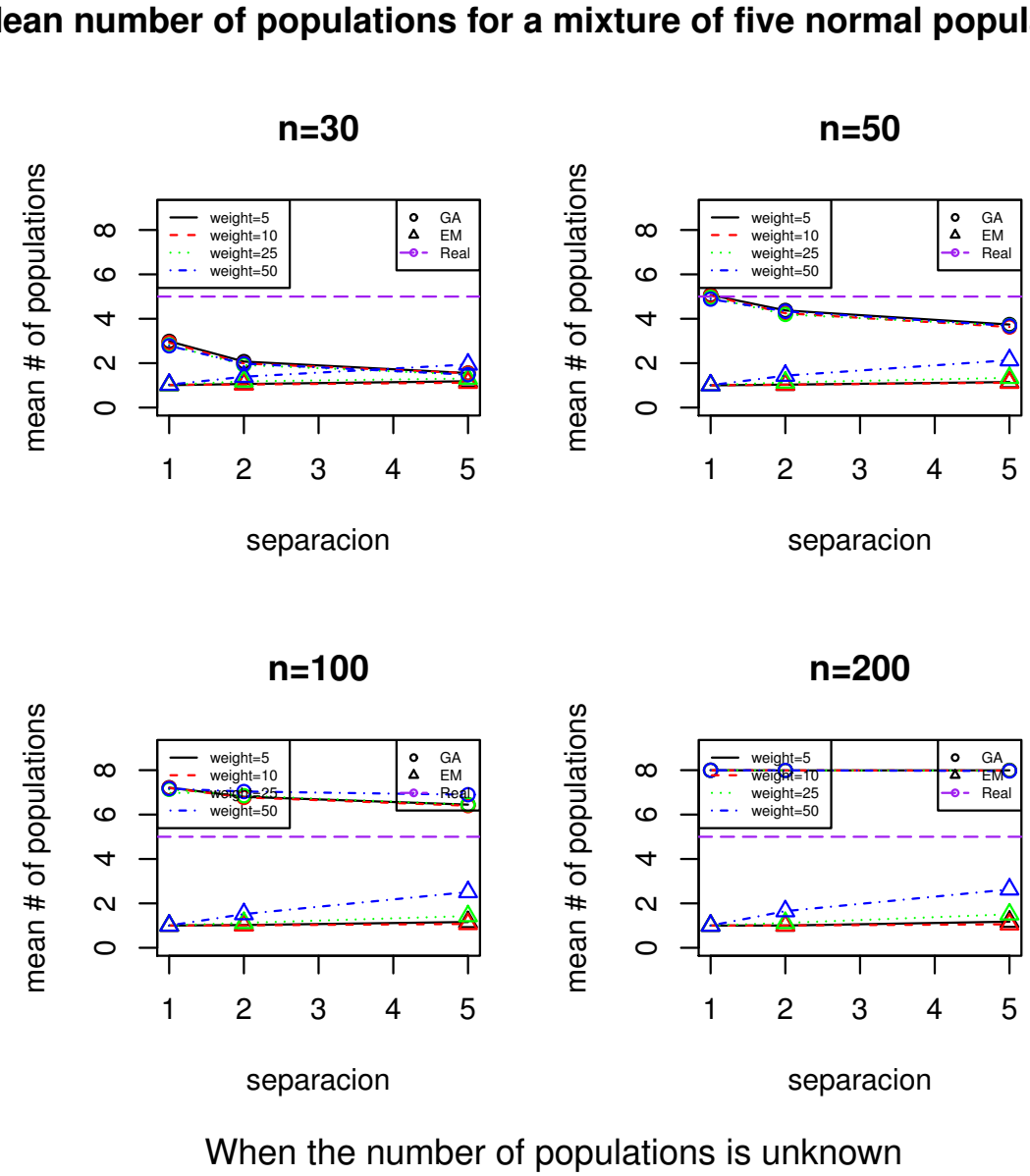

Figure 4-15.: Mean estimated number of populations for the mixture of five normal distributions. Source: Build by the authors

Two known populations - Part 1/2				*Number of populations for Gamma Distribution*			
	Parameters			**Mean**		**Standard Deviation**	
#	**Sample Size**	**separation**	**weight**	**GA**	**EM**	**GA**	**EM**
1	30	1	5	2,480	1,009	0,615	0,097
2	50	1	5	3,276	1,004	0,514	0,063
3	100	1	5	4,936	1,000	0,245	0,000
4	200	1	5	5,000	1,000	0,000	0,000
5	30	2	5	2,392	1,004	0,592	0,063
6	50	2	5	3,238	1,002	0,479	0,045
7	100	2	5	4,932	1,000	0,253	0,000
8	200	2	5	5,000	1,000	0,000	0,000
9	30	5	5	2,304	1,010	0,580	0,100
10	50	5	5	3,214	1,004	0,465	0,063
11	100	5	5	4,938	1,000	0,241	0,000
12	200	5	5	5,000	1,000	0,000	0,000
13	30	1	10	2,478	1,027	0,643	0,263
14	50	1	10	3,292	1,006	0,521	0,077
15	100	1	10	4,958	1,000	0,201	0,000
16	200	1	10	5,000	1,000	0,000	0,000
17	30	2	10	2,396	1,016	0,600	0,126
18	50	2	10	3,298	1,004	0,488	0,063
19	100	2	10	4,930	1,000	0,255	0,000
20	200	2	10	5,000	1,000	0,000	0,000
21	30	5	10	2,340	1,018	0,581	0,133
22	50	5	10	3,270	1,010	0,508	0,100
23	100	5	10	4,950	1,000	0,218	0,000
24	200	5	10	5,000	1,000	0,000	0,000

Table 4-21.: Total populations calculated for a mixture of two gamma distributions, part 1. The second part is in Table**4-22** in page 60. Source: Build by the authors

4.2.4. Mixture of two gamma distributions

The results of the simulations of the experiments for the estimation of populations of the mixture of two normal gamma distributions can be observed in Tables **4-21** and **4-22** and their graphic in Figure **4-16**, in page 61. As an initial value, a start value of 5 populations was set. Similar to the results from the mixture of normal distributions, the GA tend to overestimate the number of populations, when the sample size was large enough to compute all the populations, but this was done with a lot of dispersion, as it can be seen in the Tables. For the EM algorithm it can be seen that in almost all the cases it underestimated the number of populations, even with big separation, sample size and weight, but the dispersion in almost all cases was 3 times smaller that the one achieved with GA. The conclusion is that neither of the methods had the accuracy needed to estimate correctly the number of populations for a mixture of two gamma distributions.

Two known populations - Part 2/2				*Number of populations for Gamma Distribution*				
	Parameters			**Mean**		**SD**		
#	**Sample Size**	**separation**	**weight**	**GA**	**EM**	**GA**	**EM**	
25	30	1	25	2,571	1,065	0,649		0,306
26	50	1	25	3,340	1,014	0,549		0,118
27	100	1	25	4,945	1,000	0,229		0,000
28	200	1	25	5,000	1,000	0,000		0,000
29	30	2	25	2,514	1,028	0,659		0,165
30	50	2	25	3,399	1,013	0,563		0,112
31	100	2	25	4,940	1,002	0,238		0,045
32	200	2	25	5,000	1,000	0,000		0,000
33	30	5	25	2,440	1,026	0,609		0,171
34	50	5	25	3,230	1,012	0,464		0,111
35	100	5	25	4,944	1,002	0,239		0,045
36	200	5	25	5,000	1,000	0,000		0,000
37	30	1	50	2,737	1,100	0,653		0,447
38	50	1	50	3,514	1,054	0,675		0,280
39	100	1	50	4,938	1,002	0,241		0,045
40	200	1	50	5,000	1,000	0,000		0,000
41	30	2	50	2,827	1,096	0,794		0,296
42	50	2	50	3,593	1,037	0,651		0,201
43	100	2	50	4,963	1,000	0,190		0,000
44	200	2	50	5,000	1,000	0,000		0,000
45	30	5	50	2,556	1,056	0,691		0,231
46	50	5	50	3,480	1,066	0,644		0,249
47	100	5	50	4,944	1,000	0,230		0,000
48	200	5	50	5,000	1,000	0,000		0,000

Table 4-22.: Total populations calculated for a mixture of two gamma distributions, part 2. The first part is in Table**4-21** in page 59. Source: Build by the authors

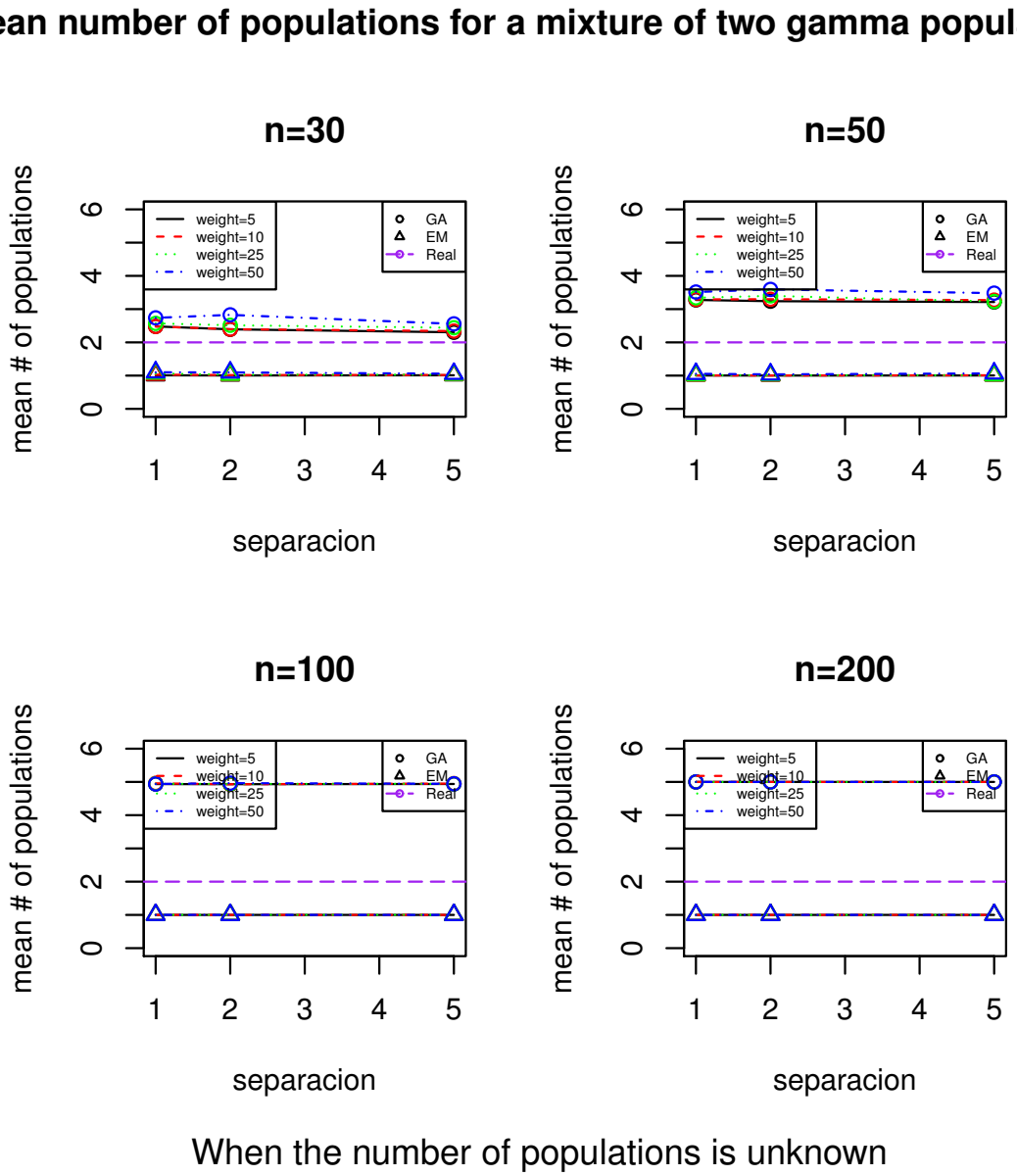

Figure 4-16.: Mean estimated number of populations for the mixture of two gamma distributions. Source: Build by the authors

4.2.5. Mixture of three gamma distributions

The mean number of populations estimated using GA and EM algorithm for a mixture of three gamma distribution for the 48 scenarios used in this research can be found on Tables **4-23** and **4-23** and their graphic in Figure **4-17**, in page 65. The starting values for the estimation was 6 populations for both methods. The estimation of the number of populations in this case was very difficult because of the numerical stability due to the calculus of the maximum likelihood for different sets of gamma distributions, and this was more noticeable with largue set of data and largue separation, for both methods. For the values that could be computed, the EM algorithm was more precise, having lower standard deviation, but in all cases the detection of more of one distribution was small, because of the mean number of population, that is in all cases, very close to one. For the GA the results were different, because of the initial value, when the sample size allows for, the estimation was the same as the 6 populations, for the other cases when there was not enough data, the number of populations was close. As a conclusion, in this scenario, neither of the methods had satisfactory performance, because none of them could calculate the real number of populations.

Three known populations - Part 1/2				*Number of populations for Gamma Distribution*				
		Parameters		**Mean**		**SD**		
	n	**separacion**	**weight**	**GA**	**EM**	**GA**	**EM**	
1	30	1	5	2,524	1,095	0,773		0,370
2	50	1	5	4,054	1,080	0,680		0,273
3	100	1	5	5,640	1,010	0,540		0,100
4	200	1	5	6,000	1,000	0,000		0,000
5	30	2	5	2,444	1,150	0,616		0,366
6	50	2	5	4,079	1,025	0,587		0,158
7	100	2	5	5,660	1,010	0,525		0,100
8	200	2	5	6,000	1,000	0,000		0,000
9	30	5	5	2,276	1,034	0,790		0,183
10	50	5	5	3,850	1,028	0,748		0,205
11	100	5	5	5,535	1,000	0,592		0,000
12	200	5	5	6,000	1,000	0,000		0,000
13	30	1	10	2,550	1,146	0,783		0,422
14	50	1	10	4,021	1,072	0,649		0,260
15	100	1	10	5,670	1,005	0,492		0,071
16	200	1	10	6,000	1,000	0,000		0,000
17	30	2	10	2,433	1,097	0,626		0,396
18	50	2	10	4,042	1,040	0,624		0,200
19	100	2	10	5,630	1,005	0,504		0,071
20	200	2	10	6,000	1,000	0,000		0,000
21	30	5	10	2,600	1,270	0,736		0,693
22	50	5	10	3,938	1,015	0,664		0,124
23	100	5	10	5,555	1,005	0,573		0,071
24	200	5	10	6,000	1,000	0,000		0,000

Table 4-23.: Total populations calculated for a mixture of three gamma distributions, part 1. The second part is in Table**4-24** in page 64. Source: Build by the authors

Three known populations - Part 2/2				*Number of populations for Gamma Distribution*				
		Parameters		**Mean**		**SD**		
	n	**separacion**	**weight**	**GA**	**EM**	**GA**	**EM**	
25	30	1	25	2,000	1,222	0,756		0,441
26	50	1	25	4,200	1,182	0,696		0,664
27	100	1	25	5,747	1,000	0,437		0,000
28	200	1	25	6,000	1,000	0,000		0,000
29	30	2	25	2,625	1,222	0,518		0,441
30	50	2	25	3,889	1,000	0,928		0,000
31	100	2	25	5,698	1,011	0,487		0,107
32	200	2	25	6,000	1,000	0,000		0,000
33	30	5	25	2,818	1,545	0,874		0,688
34	50	5	25	4,500	1,000	0,707		0,000
35	100	5	25	5,621	1,034	0,511		0,237
36	200	5	25	6,000	1,000	0,000		0,000
37	30	1	50	2,000	1,250	0,816		0,463
38	50	1	50	NA	1,000	NA		NA
39	100	1	50	NA	NA	NA		NA
40	200	1	50	NA	NA	NA		NA
41	30	2	50	2,500	1,000	0,577		0,000
42	50	2	50	4,000	1,000	0,000		0,000
43	100	2	50	NA	NA	NA		NA
44	200	2	50	NA	NA	NA		NA
45	30	5	50	NA	NA	NA		NA
46	50	5	50	NA	NA	NA		NA
47	100	5	50	NA	NA	NA		NA
48	200	5	50	NA	NA	NA		NA

Table 4-24.: Total populations calculated for a mixture of three gamma distributions, part 2. The first part is in Table**4-23** in page 63. Source: Build by the authors

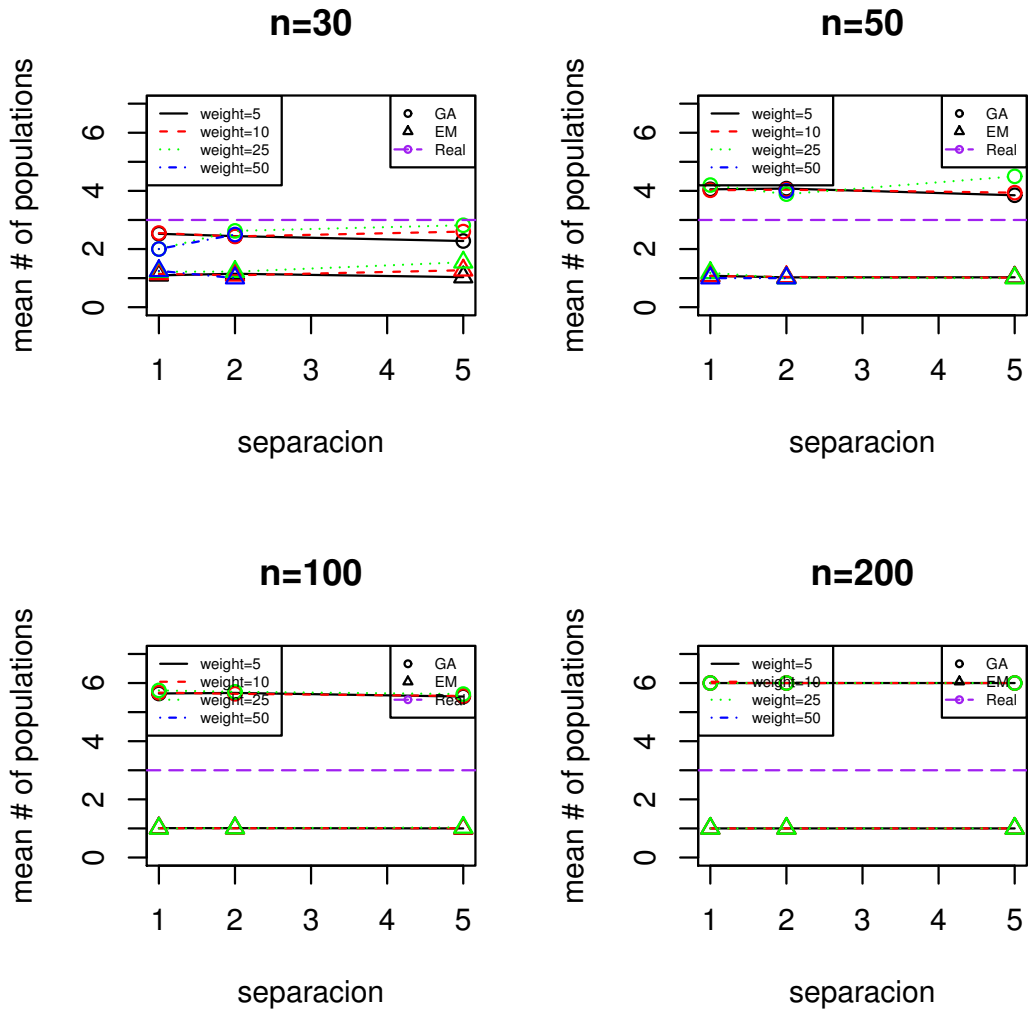

Figure 4-17.: Mean estimated number of populations for the mixture of three gamma distributions. Source: Build by the authors

4.2.6. Mixture of five gamma distributions

The results of the estimation of the number of populations for a mixture of 5 gamma populations can be seen in Tables **4-25** and **4-26** and their graphic in Figure **4-18**, in page 69, respectively. The number of initial populations was set to 8, and this carried a huge numerical instability, for this reason the GA could not work, generating errors from the first simulation.The EM algorithm could detect the number of populations in all cases, because it did not yielded NA as a result, but, as the case with 3 unknown populations in section 4.2.5, in almost all cases it could not detect more than one population, because of the proximity of the mean number of population to one. As a remark, all the scenarios had a bigger standard deviation than their analogous from 2 and 3 populations. This indicates that the algorithm was less precise. As a conclusion, the GA should not be used to detect the number of populations, and even more if there are a lot of populations, or if there are clues that these populations follows a gamma distribution. The EM algorithm was not precise to find the number of populations.

In this chapter, we described the results of the simulation, and a summary of the conclusions can be seen on chapter 6. In the next chapter there is an illustrative example of the use of this algorithm.

Five known populations - Part 1/2				*Number of populations for Gamma Distribution*				
		Parameters		**Mean**		**SD**		
#	**n**	**separacion**	**weight**	**GA**	**EM**	**GA**	**EM**	
1	30	1	5	NA	1,315	NA		0,627
2	50	1	5	NA	1,253	NA		0,646
3	100	1	5	NA	1,126	NA		0,471
4	200	1	5	NA	1,025	NA		0,157
5	30	2	5	NA	1,392	NA		0,690
6	50	2	5	NA	1,365	NA		0,785
7	100	2	5	NA	1,085	NA		0,330
8	200	2	5	NA	1,005	NA		0,071
9	30	5	5	NA	1,409	NA		0,753
10	50	5	5	NA	1,286	NA		0,595
11	100	5	5	NA	1,080	NA		0,353
12	200	5	5	NA	1,005	NA		0,071
13	30	1	10	NA	1,467	NA		0,940
14	50	1	10	NA	1,289	NA		0,632
15	100	1	10	NA	1,165	NA		0,499
16	200	1	10	NA	1,000	NA		0,000
17	30	2	10	NA	1,403	NA		0,709
18	50	2	10	NA	1,346	NA		0,705
19	100	2	10	NA	1,145	NA		0,430
20	200	2	10	NA	1,020	NA		0,140
21	30	5	10	NA	1,317	NA		0,688
22	50	5	10	NA	1,300	NA		0,712
23	100	5	10	NA	1,150	NA		0,468
24	200	5	10	NA	1,035	NA		0,184

Table 4-25.: Total populations calculated for a mixture of five gamma distributions, part 1. The second part is in Table**4-26** in page 68. Source: Build by the authors

Five known populations - Part 2/2		*Number of populations for Gamma Distribution*						
		Parameters		**Mean**		**SD**		
#	**n**	**separacion**	**weight**	**GA**	**EM**	**GA**	**EM**	
25	30	1	25	NA	1,314	NA		0,550
26	50	1	25	NA	1,371	NA		0,711
27	100	1	25	NA	1,171	NA		0,578
28	200	1	25	NA	1,040	NA		0,196
29	30	2	25	NA	1,366	NA		0,671
30	50	2	25	NA	1,465	NA		0,847
31	100	2	25	NA	1,133	NA		0,396
32	200	2	25	NA	1,035	NA		0,253
33	30	5	25	NA	1,373	NA		0,701
34	50	5	25	NA	1,391	NA		0,845
35	100	5	25	NA	1,181	NA		0,490
36	200	5	25	NA	1,030	NA		0,171
37	30	1	50	NA	1,373	NA		0,662
38	50	1	50	NA	1,243	NA		0,469
39	100	1	50	NA	1,106	NA		0,355
40	200	1	50	NA	1,050	NA		0,279
41	30	2	50	NA	1,167	NA		0,519
42	50	2	50	NA	1,237	NA		0,485
43	100	2	50	NA	1,383	NA		0,880
44	200	2	50	NA	1,180	NA		0,608
45	30	5	50	NA	1,309	NA		0,663
46	50	5	50	NA	1,440	NA		0,686
47	100	5	50	NA	1,257	NA		0,612
48	200	5	50	NA	1,162	NA		0,600

Table 4-26.: Total populations calculated for a mixture of five gamma distributions, part 2. The first part is in Table**4-25** in page 67. Source: Build by the authors

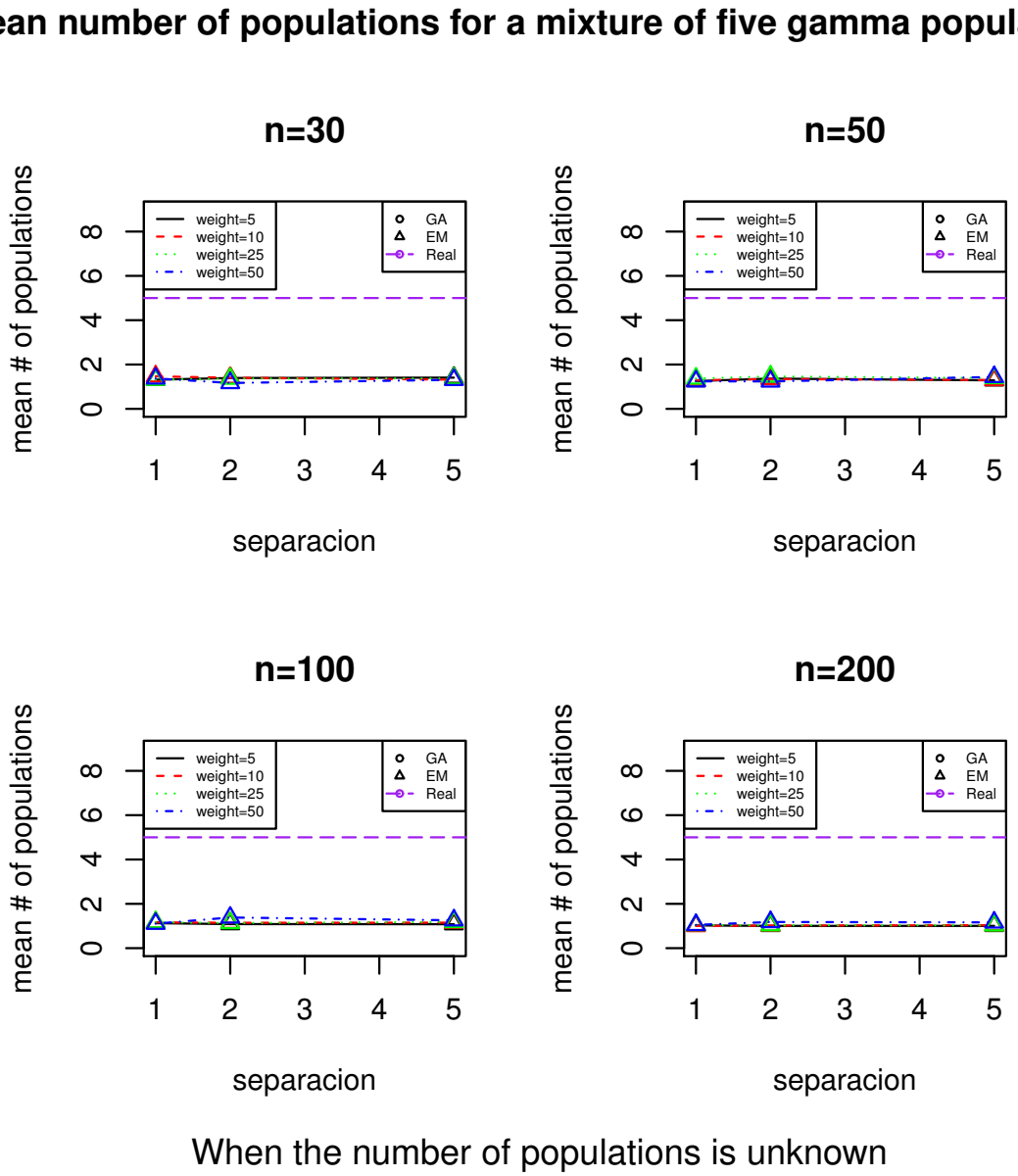

Figure 4-18.: Mean estimated number of populations for the mixture of five gamma distributions. Source: Build by the authors

5. Illustrative examples

The data for this illustration were taken from a study conducted by Estrada, *et. al.*, 1988 [12], the permission to use de data set was given of the Seguro Social. This study had as an objective to measure 69 anthropometric parameters from a workforce in Colombia. The data were taken from males and females from 20 to 60 years old, and the aim was to get a characterization of the population, and with the information taken from this database, to get design spaces and equipment for the use of the Colombian workers, because historically these have been designed using international standards or heuristically. From this study, the data on BMI (Body Mass Index) have been selected as the variable to analyze, because of the importance to describe the body and therefore the designs to do for the colombian workers, also is a variable that is important to show the risk of mortality by circulatory diseases or cancer [12]. The histogram and the density can be seen in Figure **5-1** where it shows a form of a bell, but with a heavy tail on the right, and a little hump around a BMI of 30. Looking very carefully, it can be seen another humps on 24 and 28. For this reason this might not follow a normal distribution, but this is checked with graphical and numerical analysis. The QQ plot is shown in Figure **5-2**. In the QQ plot it is shown that the distribution has heavy tails, this weakens the assuption of having only one normal distribution. The numerical test is made using the Kolmogorov–Smirnov test, for a two sided hypotesis. We observed a $p - value < 2{,}20x * 10^{-16}$, and this analysis confirms that the distribution is not a normal one. For this reason, an analysis using a mixture of distributions could be appropriate.

The first step to analyze the data is to find the number of populations. This was achieved using the algorithms evaluated in this research. Because of the results of the simulations in Chapter 4 shows that the EM and GA could not estimate the number of populations correctly, the proposed strategy to follow is to set different scenarios and compare the results of EM and GA with the estimated for the data using the kernel density estimates, the initial parameter for the number of populations was set at 4, the GA was set for a population size of 500 and 500 iterations, the EM algorithm was set of 500 iterations, and the amount of data to analyze was 2100, looking for a better convergence, due to the results explained in the previous chapter.

The analysis was later conducted using the same algorithms for the simulations, and the parameters estimated are shown in table **5-1**. The parameters of every populations were different from the calculations of both methods, and as a way to assess the adjustment to the data, the graph **5-3** was created. In this graphic, the best method for this data set is the EM algorithm with 3 populations, because it is the one that looks closer to the estimated

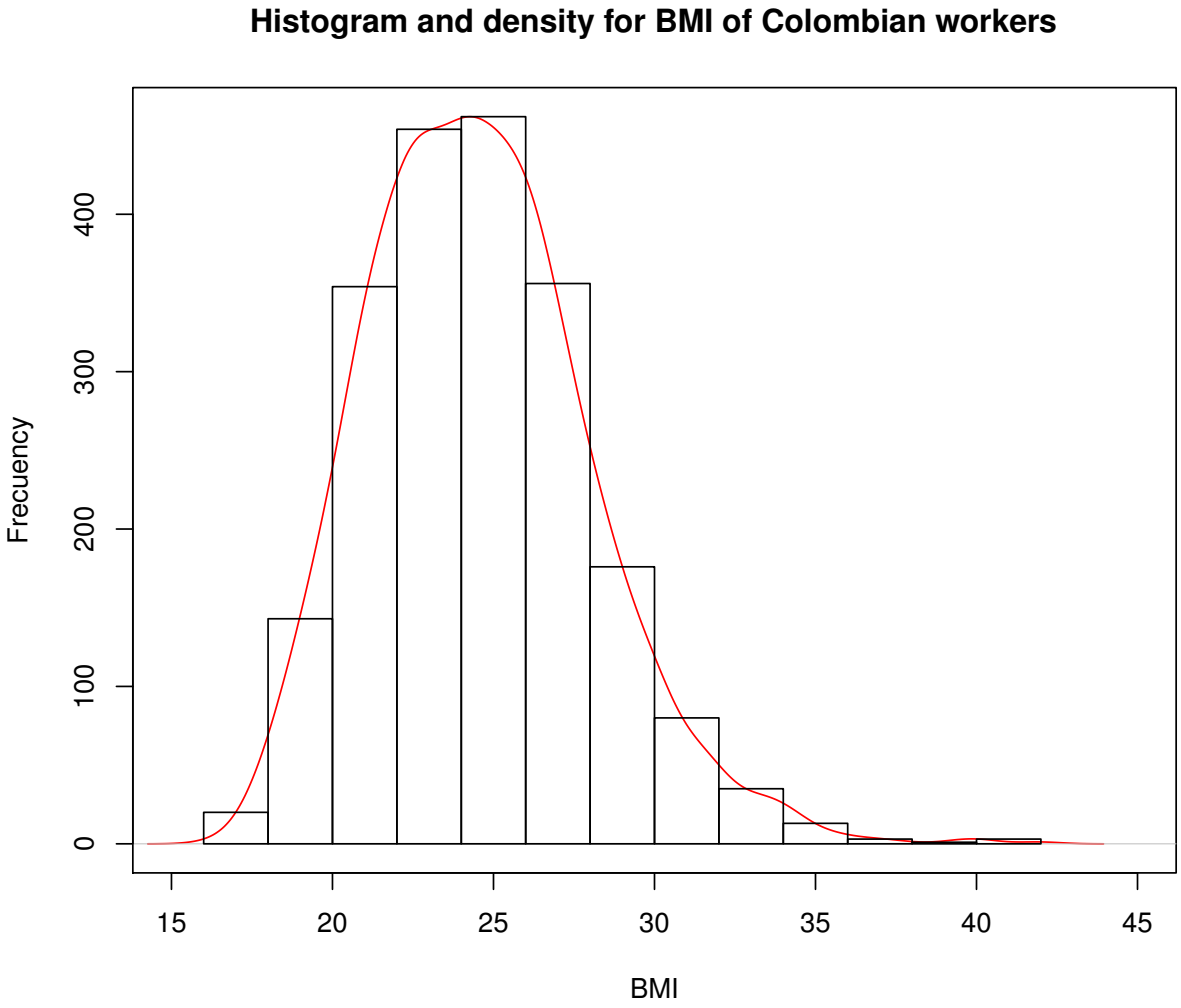

Figure 5-1.: Histogram and estimated density for data about the BMI of colombian workers

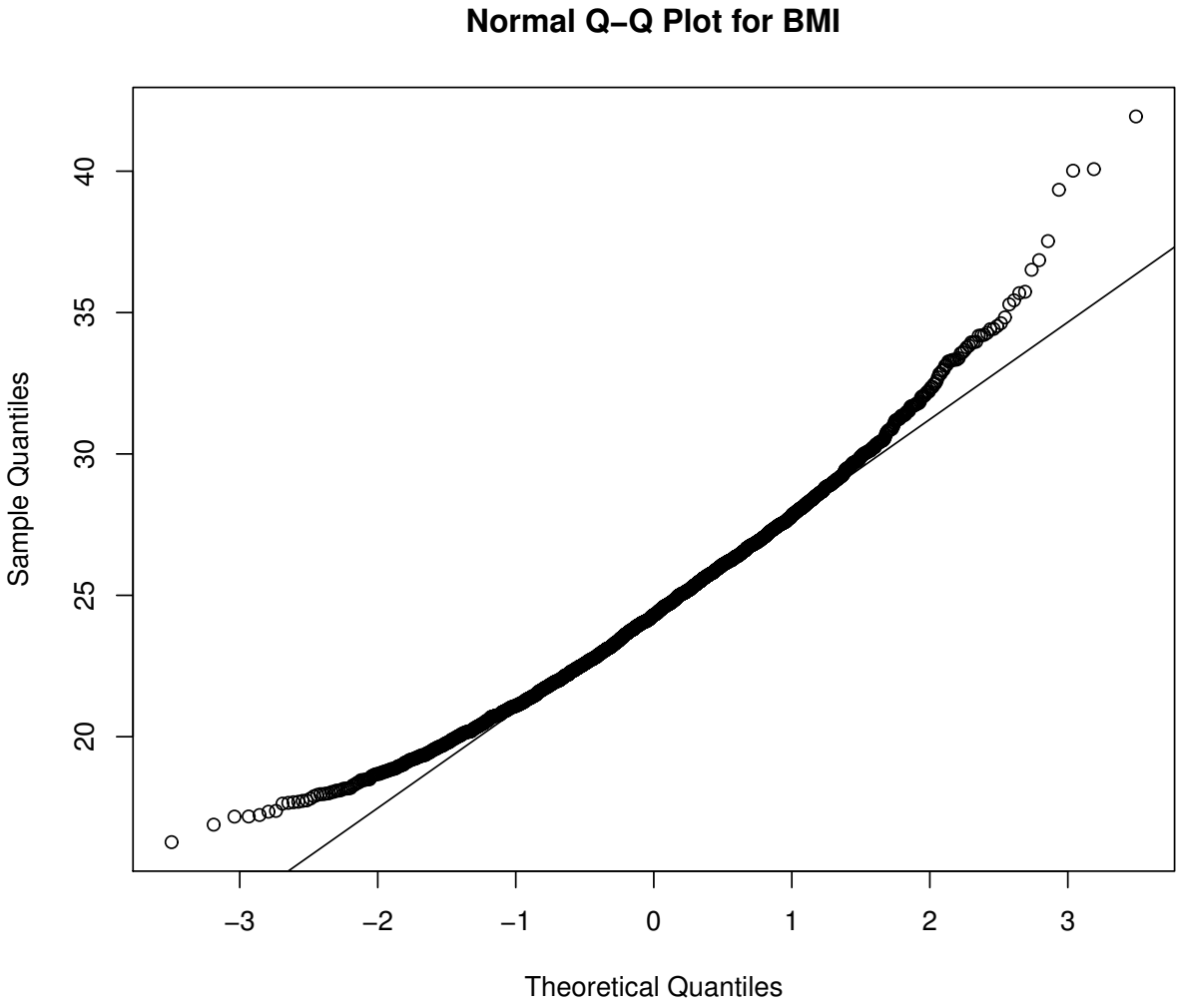

Figure 5-2.: QQ plot the BMI of colombian workers

Number of populations	Pop number	GA			EM		
		π	μ	σ	π	μ	σ
2	1	0.4567	26.3176	3.6591	0.7938	23.7285	2.7923
	2	0.5433	23.0441	2.3605	0.2062	27.6599	3.8662
3	1	0.2657	23.0418	2.3409	0.2381	21.1638	1.6954
	2	0.4681	25.9696	3.9413	0.1191	28.7908	3.9641
	3	0.2662	23.5178	2.1600	0.6429	25.0014	2.5691
4	1	0.1590	23.2227	2.6489	0.2934	21.3270	1.7535
	2	0.3252	26.9433	3.7198	0.0065	24.0586	0.0294
	3	0.3662	23.1989	2.4640	0.5700	25.2685	2.4292
	4	0.1495	23.9916	2.7161	0.1302	28.6128	3.9589

Table 5-1.: Parameters estimated from the mixture of BMI of colombian workers

density .This method gives the information to conclude that there might be three groups of Colombian workers, one with a the 24 % of the people, with a healthy BMI, with mean 21, the majority 64 % with overweight with a BMI of 25, and with a standard deviation of 2,6, and the last one with a 12 % of people, with a BMI of 28, close to the obesity.
As a conclusion, the methods can be used for real case studies with results that can describe the data. As a recommendation, we endorse further studies of the number of populations, because it is a critical input and the methods here exposed are not very accurate for the estimation of the number of populations in the mixture. We recommend to follow the EM algorithm for the estimation of the number of populations, and next using an evolutive algorithm if the distribution is not a mixture of normal populations, also, to implement a test to define the family of distributions in the mixture, to use the Hellinger Distance to compare the results with the Kernel Density.

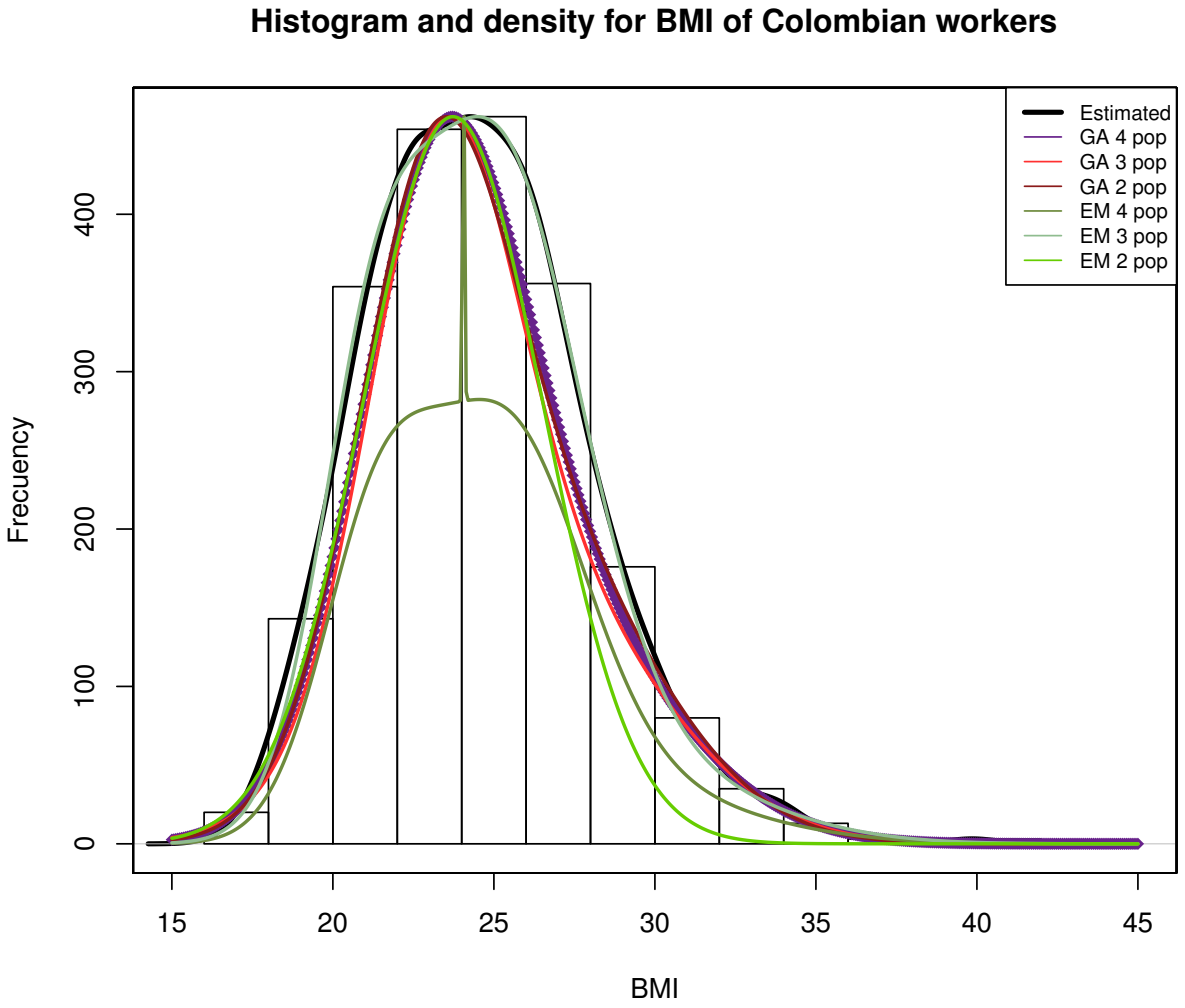

Figure 5-3.: Comparison of method to estimate the mixture of BMI of colombian workers

6. Conclusions

A comparison between evolutive algorithms and traditional methods for the estimation of the parameters of mixture models was conducted in this study. The evolutive algorithm was represented by the Genetic Algorithm (GA) and the traditional method by the Expectation – Maximization Algorithm (EM); the parameters evaluated were:whether the number of populations is known or unknown, type distribution, number and weight of populations, sample size and separation between means of the populations in the mixture, for the case of the normal mixture, or the α parameter in a gamma mixture. This study was made using simulations on `R` [26] with a code proposed for this research, and then the Hellinger distance was calculated to make the comparison. When the simulations were running, the first thing to notice is that the EM simulation's running time was significantly higher than the GA simulation's running time; the Hellinger distance is smaller when the sample increases, and the composition of the mixture is even.

For a mixture of normal distributions when the number of populations is known, the results obtained using the EM were better than the ones achieved with GA, because in general it had smaller distance and standard deviation. For the GA, this method needs sample sizes of at least 50 items or the population to have a weight of at least 10 % to have the sensibility to detect all the populations; when the amount of populations increase, so does the number of data needed to estimate the parameters. The separation between means have an opposite effect in the distance, the EM has lower distance as the distance increases. For the reasons exposed, the EM is better at the estimation of the parameters in a mixture of normal distributions than the GA.

For a mixture of gamma distributions, the methods have lower distance when there are large samples, and the weights in the mixture are even.In this case, the GA could not detect all the populations when at least one of the populations had weight equal or less than 10 % and the sample size was small, less than 100 units. For the mixture of gamma distributions, the GA had better results when the separation was small, 1 or two β. One problem with both methods was numerical instability that was more noticeable as the number of populations increased, and there was a lack of convergence in the results of the EM algorithm. For that reason, the GA is a better option to estimate the parameters in this mixture.

In the cases when the number of populations is unknown for a mixture of normal distributions neither of the methods, GA nor GAP analysis are exact, but the EM algorithm is is closer to the real value of two populations, the algorithm can compute all the populations, as seen in the results of the estimation of the parameters with the number of populations known.

For three populations, as the number of population increases, the performance of the GA for the estimation of the number of populations in a mixture decreases, the GAP analysis had the smallest variation in all the scenarios, but it is not exact.. In the case when there are five populations, for the GAP analysis, in all the cases the results were the number of populations was underestimated, the only close was when a population had the 50 % of the data and segregation of 5 standard deviation, for the rest, it could not even notice that there was a mixture. In conclusion, neither of the methods could correctly estimate the number when there was a big amount of populations.

For the number of populations unknown in a mixture of gamma distributions, neither of the methods could estimate correctly the number of populations. As the number of population increases, the GA starts to present numerical instability, being unable to estimate the parameters. As a conclusion, the GA should not be used to detect the number of populations, and even more if there are a lot of populations, or if there are clues that these populations follows a gamma distribution.

The methods were used in a real problem, to estimate the number of populations of Colombian workers based on their Body Mass Index. As a conclusion, the methods can be used in real applications with results that can describe the data. The recommendation is to study the number of populations, because it is a critical input and the methods here exposed are not very accurate.

The biggest problem found in this study, was the time to process the GA simulations, one recommendation is to improve its performance. Another recommendation to future studies is to explore the behavior of the estimation of the number of populations with larger sample size, more iterations, evaluating the effect of heterogeneous variance in the mixture and using other distributions.

A. Appendix: R Packages

A.1. Packages for Genetic Algorithms

A.1.1. gafit

This package was developed in 2002 and uses the genetic algorithm for minimizing the value of a given function. This package is very simple, as it has only one function, with minimum input parameters: function to minimize, the start values for the chromosomes and number of iterations. This function also includes a thermal and step between samples, and use this term are not common on the Genetic Algorithm literature, and given that this package was developed 13 years ago, and it has some documented bugs, the decision was not to use it on this investigation.

A.1.2. galts

This package is the Genetic algorithms and C-steps based LTS (Least Trimmed Squares) estimation. This package is useful to estimate a function without being affected in the presence of outliers. With galts it is possible to detect regression outliers. The estimation function, using genetic algorithms adjusted a set of data to a formula, and gives as a result the coefficients, LTS criterion and the method used for the optimization. For this nature, this package is not useful for this research.

A.1.3. mcga

The mcga package was developed in march, 2014 and reviewed in february, 2015. It is a very complete set for using genetic algorithms as a method to solve a problem. It is used to solve real valued optimization problems. It requieres as inputs the population size, number of parameters to estimate, genetic algorithm parameters as crossover and mutation probability, elitism, the search space, and maximum iterations. Gives as a result the population and the value for every chromosome. This package also allows optimizing a multi objective function.

A.1.4. rgenoud

This package has more than 5 versions so far. For the optimization problem this package uses a combination of evolutive and Newton methods. This can be used when the function to

optimize does not have a derivative. This method can use several processes at a time. This package only has one function, genound, for the GENetic Optimization Using Derivatives. This function has a wide pool of parameters to control the optimization, including some set for the performance of the computation, for example, it can control the RAM used. The reason to not use this package for this research was the combination with Newton methods, as we wanted to evaluate the performance of the pure evolutive algorithm.

A.1.5. genalg

This package has been released on march 16, 2015 and it has the possibility to optimize a set of real valued and binary data, along with the graphics of the evolution of the populations and a complete summary of the parameters found and the performance of the algorithm. The functions of this package use as input the function to evaluate. This function can only receive the chromosome as an input. Other parameters are the elitism, mutation chance, population size and the search space.

A.1.6. GA

This package has a wide pool of functions to optimize a problem using genetic algorithms. This package also has the advantage of running in parallel on two or more processors. The command `ga` allows to optimize an objective function, giving the advantage of allowing that function to receive the chromosome and other parameters for the optimization. This makes the package more flexible than the others evaluated in this investigation. The package can be used with real valued, binary and permutation. Compared with other packages, this one possesses a wider pool of parameters for the improve of the estimation, such as seed, running in parallel, keep the best chromosome, a specific function to compute the crossing over and selection. The return of the optimization is an object with all the data obtained in the optimization. This is a very complete package of genetic algorithms for the improvement of functions, but the main issue is the performance, because it was three times longer to optimize than `genalg`.

A.2. Packages for mixture models

A.2.1. mclust

This package uses the EM algorithm for model based clustering, classification and density estimation. This package offers a lot of different functions to fit the data, compute the cdf, errors, scatter plot, density estimations for each point, the number of clusters, information criterion, and some datasets from some cases of studies, such as thyroid function and the

acidity of lakes in North America. This package is used in this study to assess the number of populations using EM algorithms because of the usability.

A.2.2. BayesMix

This package uses a Bayesian framework for fitting mixture models of univariate Gaussian distributions. This is a new package, being released on July 2015 and allows to represent the data using plots of the data and their posteriori density, using JAGS (Just Another Gibbs Sampler) for analysis of Bayesian hierarchical models. This package was not used because in this study, we focus on the EM algorithm as the illustration of traditional methods for the estimation of mixture models.

A.2.3. Rmixmod

This package, which was released on march, 2014, is developed for the Supervised and unsupervised classification with mixture modeling. This package is very complete for a mixture of Gaussian or multinomial mixture models, giving tools to plot, define parameters, find and analyze clusters, and some data to run the models. This one was not used in this study because they do not support the mixture of other distributions, such as gamma mixture.

A.2.4. mixtools

This package was released in 2015, It contains a wide collection of functions to analyze finite mixture models. This package allows to compute CDF, densities, bootstraps for the calculation of the likelihood, plot the mixture and allows to perform simulations of mixtures. This package supports a mixture of multiple normal distributions, logistic regressions, multinomials, gamma distributions using the EM algorithm. The pool of mixtures of distributions that this package allows to work with is the reason to choose this package over others, also, It was tested and it is stable and has a good performance.

A.2.5. Flexmix

This package, uses a general framework for finite mixtures of regression models. They use the EM algorithm to do so. This package allows to compute information criteria, it has examples of applications, making it easier to understand allows to cluster Gaussian distributions this can use the maximum likelihood of a wide family of distributions as Gaussian, binomial, Poisson and Gamma.

B. Appendix: Algorithm

B.1. Algorithm for number of populations known

```
EM.results= function(datos1,k, iter, expnumber){ tryCatch(
  normalmixEM (datos1, lambda = NULL, mu = NULL, sigma = NULL, k = k,
               sd.constr = NULL,
               epsilon = 1e-08, maxit = 500, maxrestarts=20,
               verb = FALSE, fast=FALSE, ECM = FALSE,
               arbmean = TRUE, arbvar = TRUE),

  warning = function(w) {
    list(lambda=rep(NA,k),mu=rep(NA,k), sigma=rep(NA,k))
  },
  error = function(e) {list(lambda=rep(NA,k),mu=rep(NA,k), sigma=rep(NA,k))}
  )
  }

GAestimationk<-function(cromosoma){



  cromosoma=t(round(cromosoma, digits=0))

  datosGA = datos

  datosGak = matrix(nrow=nrow(datos),ncol=(ncol(datos)+1))

  for (i in 1:3) {datosGak[,i]=datos[,i]}

  datosGak[,4]=t(cromosoma)

  colnames(datosGak)=c("Observation","Value","RealPopulation","Chromosome")


  ParameterValue=DataSettings[,,expnumber]



  tablacromosoma=table(cromosoma)
```

```
  Results=matrix(NA,nrow=max(cromosoma), ncol=6)

  cuenta_cromosoma=rep(NA,max(cromosoma))

  for (h in 1:max(cromosoma)) {cuenta_cromosoma[h]=sum(cromosoma==h)}

 for (j in 1:max(cromosoma)){

   if (cuenta_cromosoma[j]>3){

     Population=datosGak[cromosoma==j,3]

     Param=fitdistr(Population, densfun="normal")

     Results[j,1]=ParameterValue[j,1]

     Results[j,2]=coef(Param)[1] #parameter 1 (mu or alpha)

     Results[j,3]=coef(Param)[2] # parameter 2 (sigma or beta)

     Results[j,4]=abs(ParameterValue[j,2]-(cuenta_cromosoma[j]/
       sum(cuenta_cromosoma)))/ParameterValue[j,2]

     Results[j,5]=abs(ParameterValue[j,3]-coef(Param)[1])/
       ParameterValue[j,3]

     Results[j,6]=abs(ParameterValue[j,4]-coef(Param)[2])/
       ParameterValue[j,4]

   }

 }

                  meanerror=mean(Results[cuenta_cromosoma>3,c(4,5,6)])


      return(meanerror)
}

GAestimationk_full<-function(cromosoma){


  cromosoma=t(round(cromosoma, digits=0))

  datosGA = datos

  datosGak = matrix(nrow=nrow(datos),ncol=(ncol(datos)+1))
```

```
for (i in 1:3) {datosGak[,i]=datos[,i]}

datosGak[,4]=t(cromosoma)

colnames(datosGak)=c("Observation","Value","RealPopulation","Chromosome")


tablacromosoma=as.numeric(table(cromosoma))


Results=matrix(NA,nrow=max(cromosoma), ncol=7)


cuenta_cromosoma=rep(NA,max(cromosoma))

for (h in 1:max(cromosoma)) {cuenta_cromosoma[h]=sum(cromosoma==h)}

for (j in 1:max(cromosoma)){

  if (cuenta_cromosoma[j]>3){

Population=datosGak[cromosoma==j,3]

Param=fitdistr(Population, densfun="normal")

Results[j,1]=(cuenta_cromosoma[j])/sum(cuenta_cromosoma)

Results[j,2]=coef(Param)[1] #parameter 1 (mu or alpha)

Results[j,3]=coef(Param)[2] # parameter 2 (sigma or beta)
  Results[j,4]=abs(ParameterValue[j,2]-(cuenta_cromosoma[j]/
      sum(cuenta_cromosoma)))/ParameterValue[j,2]

Results[j,5]=abs(ParameterValue[j,3]-coef(Param)[1])/ParameterValue[j,3]

Results[j,6]=abs(ParameterValue[j,4]-coef(Param)[2])/ParameterValue[j,4]

  }

}



          meanerror=mean(Results[cuenta_cromosoma>3,c(4,5,6)])

          Results[,7]=as.numeric(table(t(cromosoma)==datos[,2]))[2]/
              sum(cuenta_cromosoma)
```

```
      return(Results)
}

GenerateData<-function(n1,DataSettings){

CalculatedWeights=rmultinom(1,
    size=n1, prob=DataSettings[,2])

GeneratedData=c(rnorm(CalculatedWeights[1,1],
    mean=DataSettings[1,3], sd=DataSettings[1,4]))

PopID=c(rep(1,CalculatedWeights[1,1]))

for (i in 2:k) {
  GeneratedData=c(GeneratedData,rnorm(CalculatedWeights[i,1],
    mean=DataSettings[i,3],sd=DataSettings[i,4]))
  PopID=c(PopID,rep(i,CalculatedWeights[i,1]))
}

Seq=seq(1,n1)
Datos=cbind(Seq,PopID,GeneratedData)

rand<-sample(nrow(Datos))



Desordenados=Datos[rand,]

return(Desordenados)
}


library(MASS) #For computing the likelihood and fitting the distribution

library(genalg) #for the evaluation using GA

library(mixtools) #For the evaluation using EM

library(mclust) #For the clustering of the data

library(lga) #For computing the bic


monitor <- function(obj) {
```

```
  minEval = min(obj$evaluations);

  plot(obj, type="hist");

}



k=2
max_expnumber=4
Parameters=matrix(ncol=2,nrow=max_expnumber) #n & iter
Parameters[,1]=c(30, 50, 100, 200)
Parameters[,2]=c(rep(1000,max_expnumber))

DataSettings= array(dim=c(k,4,max_expnumber))

#population number
DataSettings[,1,]=c(1, 2, 1, 2, 1, 2, 1,2)

#population weight
DataSettings[,2,]=c(0.05, 0.95, 0.05, 0.95,
    0.05, 0.95, 0.05, 0.95)

#parameter 1
DataSettings[,3,]=c(12, 13, 12, 13, 12, 13, 12, 13)

#Parameter 2
DataSettings[,4,]=c(1, 1, 1, 1, 1, 1, 1, 1)



ResultsCaso1=array(NA,dim=c(k,16,
    max(Parameters[,2]),max_expnumber))


for (expnumber in 1:max_expnumber){

  n1=Parameters[expnumber,1]


  iter=Parameters[expnumber,2]

  for (m in 1:iter){ ##poner iter


      set.seed=1+m
```

```
datos<<-GenerateData(n1,DataSettings[,,expnumber])



GAlresults = rbga(stringMin=c(rep(1,n1)), stringMax=c(rep(k,n1)),
      suggestions=NULL,
      popSize=200, iters=100,
      mutationChance=0.05,
      elitism=NA,
      evalFunc=GAestimationk,
      showSettings=FALSE, verbose=FALSE)



GAl.results<-summary(GAlresults, echo=TRUE)



  separar<-strsplit(GAl.results, "Best Solution : ")

  separar2<-unlist(strsplit(separar[[1]][2]," "))

  mejor.cromosoma<-round(as.numeric(separar2[-length(separar2)]))

 Results.GA=GAestimationk_full(mejor.cromosoma)

EM.result= EM.results(datos[,3],k,iter,expnumber)



 #Generate the results
 for (i in 1:k) {
   ResultsCaso1[i,1,m,expnumber]=DataSettings[i,2,expnumber]
   ResultsCaso1[i,2,m,expnumber]=DataSettings[i,3,expnumber]
   ResultsCaso1[i,3,m,expnumber]=DataSettings[i,4,expnumber]
   ResultsCaso1[i,4,m,expnumber]=Results.GA[i,1]
   ResultsCaso1[i,5,m,expnumber]=Results.GA[i,2]
   ResultsCaso1[i,6,m,expnumber]=Results.GA[i,3]
   ResultsCaso1[i,7,m,expnumber]=Results.GA[i,4]
   ResultsCaso1[i,8,m,expnumber]=Results.GA[i,5]
   ResultsCaso1[i,9,m,expnumber]=Results.GA[i,6]
   ResultsCaso1[i,10,m,expnumber]=Results.GA[i,7]
   #Parameters using EM algorithm
   ResultsCaso1[i,11,m,expnumber]=EM.result$lambda[i]
```

```
        ResultsCaso1[i,12,m,expnumber]=EM.result$mu[i]
        ResultsCaso1[i,13,m,expnumber]=EM.result$sigma[i]
        ResultsCaso1[i,14,m,expnumber]=abs(EM.result$lambda[i]-
            DataSettings[i,2,expnumber])/DataSettings[i,2,expnumber]
        ResultsCaso1[i,15,m,expnumber]=abs(EM.result$mu[i]-
            DataSettings[i,3,expnumber])/DataSettings[i,3,expnumber]
        ResultsCaso1[i,16,m,expnumber]=abs(EM.result$sigma[i]-
            DataSettings[i,4,expnumber])/DataSettings[i,4,expnumber]

        }
      print(paste("iteracion ",m," del experimento", expnumber))



    }

}
```

B.2. Algorithm for number of populations unknown

```
EM.results= function(datos1,k, iter, expnumber){ tryCatch(
  normalmixEM (datos1, lambda = NULL, mu = NULL, sigma = NULL, k = k,
               sd.constr = NULL,
               epsilon = 1e-08, maxit = 100, maxrestarts=20,
               verb = FALSE, fast=FALSE, ECM = FALSE,
               arbmean = TRUE, arbvar = TRUE),

  warning = function(w) {
    list(lambda=rep(NA,k),mu=rep(NA,k), sigma=rep(NA,k))
  },
  error = function(e) {list(lambda=rep(NA,k),mu=rep(NA,k), sigma=rep(NA,k))}
  )
  }


GAestimationk<-function(cromosoma){



  cromosoma=t(round(cromosoma, digits=0))

  datosGA = datos

  datosGak = matrix(nrow=nrow(datos),ncol=(ncol(datos)+1))

  for (i in 1:3) {datosGak[,i]=datos[,i]}
```

```
datosGak[,4]=t(cromosoma)

colnames(datosGak)=c("Observation","Value","RealPopulation","Chromosome")


ParameterValue=DataSettings[,,expnumber]



tablacromosoma=table(cromosoma)

Results=matrix(NA,nrow=max(cromosoma), ncol=6)

cuenta_cromosoma=rep(NA,max(cromosoma))

for (h in 1:max(cromosoma)) {cuenta_cromosoma[h]=sum(cromosoma==h)}

                    for (j in 1:max(cromosoma)){

                      if (cuenta_cromosoma[j]>3){

                        Population=datosGak[cromosoma==j,3]

                        Param=fitdistr(Population, densfun="normal")

                        Results[j,1]=cuenta_cromosoma[j]/sum(cuenta_cromosoma)

                        Results[j,2]=coef(Param)[1] #parameter 1 (mu or alpha)

                        Results[j,3]=coef(Param)[2] # parameter 2 (sigma or beta)

                        Results[j,4]=abs(ParameterValue[j,2]-
                            (cuenta_cromosoma[j]/sum(cuenta_cromosoma)))/ParameterValue[j,2]

                        Results[j,5]=abs(ParameterValue[j,3]-
                            coef(Param)[1])/ParameterValue[j,3]

                        Results[j,6]=abs(ParameterValue[j,4]-
                            coef(Param)[2])/ParameterValue[j,4]

                      }

                    }

                    meanerror=mean(Results[cuenta_cromosoma>3,c(4,5,6)])
```

```
    return(meanerror)
}

GAestimationk_unknown<-function(cromosoma){

  cromosoma=t(round(cromosoma, digits=0))
  datosGA = datos
  datosGak = matrix(nrow=nrow(datos),ncol=(ncol(datos)+1))
  for (i in 1:3) {datosGak[,i]=datos[,i]}
  datosGak[,4]=t(cromosoma)

  tablacromosoma=as.numeric(table(cromosoma))

  Results=matrix(NA,nrow=max(cromosoma), ncol=1)
  cuenta_cromosoma=rep(NA,max(cromosoma))
  for (h in 1:max(cromosoma)) {cuenta_cromosoma[h]=sum(cromosoma==h)}
                    for (i in 1:max(cromosoma)){
                      if (cuenta_cromosoma[i]>3){
                        Population=datosGak[cromosoma==i,3]
                        Param=fitdistr(Population, densfun="normal")
                        Results[i,1]=Param$loglik

                      }
                    }
                  Resultado=-1*sum(Results[cuenta_cromosoma>3,1])
```

```
      return(Resultado)
}


Nclust= function(datos1,k){ tryCatch(
  gap(datos1, K=k, B=100),

  warning = function(w) {
    list(nclust=NA)
  },
  error = function(e) {list(nclust=NA)}
  )
  }


GenerateData<-function(n1,DataSettings){

GeneratedData=c(rnorm(CalculatedWeights[1,1],
    mean=DataSettings[1,3],sd=DataSettings[1,4]))
PopID=c(rep(1,CalculatedWeights[1,1]))

for (i in 2:k) {
  GeneratedData=c(GeneratedData,rnorm(CalculatedWeights[i,1],
    mean=DataSettings[i,3],sd=DataSettings[i,4]))
  PopID=c(PopID,rep(i,CalculatedWeights[i,1]))
}

Seq=seq(1,n1)
Datos=cbind(Seq,PopID,GeneratedData)


rand<-sample(nrow(Datos))




Desordenados=Datos[rand,]

return(Desordenados)
}



library(MASS) #For computing the likelihood and fitting the distribution

library(genalg) #for the evaluation using GA

library(mixtools) #For the evaluation using EM
```

```
library(mclust) #For the clustering of the data

library(lga) #For computing the bic

setwd("E:/Tesis/Normal_N_unknown")


source('E:/Tesis/Normal_N_unknown/GAestimationk_unknown.R')
source('E:/Tesis/Normal_N_unknown/Nclust.R')
source('E:/Tesis/Normal_N_unknown/GenerarDatos.R')



k=5

max_expnumber=8
Parameters=matrix(ncol=2,nrow=max_expnumber) #n & iter
Parameters[,1]=c(30,50,100,200,30,50,100,200)
Parameters[,2]=c(rep(500,max_expnumber))


DataSettings= array(dim=c(k,4,max_expnumber))

#population number
DataSettings[,1,]=c(1,  2,      3,      4,      5,      1,      2,      3,      4,      5,
    1,  2,      3,      4,      5,      1,      2,      3,      4,      5,      1,      2,
    1,  2,      3,      4,      5,      1,      2,      3,      4,      5,      1,      2,

#population weight
DataSettings[,2,]=c(0.05,       0.24,   0.24,   0.24,   0.23,   0.05,
    0.24,       0.24,   0.24,   0.23,   0.05,   0.24,   0.24,   0.24,
    0.23,       0.05,   0.24,   0.24,   0.24,   0.23,   0.05,   0.24,
    0.24,       0.24,   0.23,   0.05,   0.24,   0.24,   0.24,   0.23,
    0.05,       0.24,   0.24,   0.24,   0.23,   0.05,   0.24,   0.24,
    0.24,       0.23)

#parameter 1
DataSettings[,3,]=c(12, 13,     14,     15,     16,     12,     13,     14,     15,     16,
    12, 13,     14,     15,     16,     12,     13,     14,     15,     16,     12,     14,
        20,     12,     14,     16,     18,     20,     12,     14,     16,     18,     20,
        16,     18,     20)

#Parameter 2
DataSettings[,4,]=c(1,  1,      1,      1,      1,      1,      1,      1,      1,      1,
    1,  1,      1,      1,      1,      1,      1,      1,      1,      1,      1,      1,
    1,  1,      1,      1,      1,      1,      1,      1,      1,      1,      1,      1)
```

```
ResultsCaso1=array(NA,dim=c(max(Parameters[,2]), 3, max_expnumber))
a=Sys.time()

for (expnumber in 1:max_expnumber){

  n1=Parameters[expnumber,1]

  iter=Parameters[expnumber,2]

  for (m in 1:iter){ ##poner iter


      set.seed=1+m


      datos<<-GenerateData(n1,DataSettings[,,expnumber])

      ResultsCaso1[m,1,expnumber]=k


      k_em_calculated=Nclust(datos[,3],k+3)
      ResultsCaso1[m,2,expnumber]=k_em_calculated$nclust


      GAlresults = rbga(stringMin=c(rep(1,n1)), stringMax=c(rep(k+3,n1)),
          suggestions=NULL,
          popSize=200, iters=100,
          mutationChance=0.05,
          elitism=NA,
          evalFunc=GAestimationk_unknown,
          showSettings=FALSE, verbose=FALSE)



      GAl.results<-summary(GAlresults, echo=TRUE)



        separar<-strsplit(GAl.results, "Best Solution : ")

        separar2<-unlist(strsplit(separar[[1]][2]," "))

        mejor.cromosoma<-round(as.numeric(separar2[-length(separar2)]))


        cuenta_cromosoma=rep(NA,max(mejor.cromosoma))
```

```
        for (h in 1:max(mejor.cromosoma)) {cuenta_cromosoma[h]=
            sum(mejor.cromosoma==h)}

      ResultsCaso1[m,3,expnumber]=length(cuenta_cromosoma[cuenta_cromosoma>3])

      print(paste("iteracion ",m," del experimento", expnumber))

    }


}

b=Sys.time()
print(b-a)
```

References

[1] A. Lijoi, R.H. M. ; Prünster., I.: Controlling the Reinforcement in Bayesian Non-Parametric Mixture Models. En: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 64 (2007), Nr. 4, p. 715–740

[2] Adele Cutler, Olga I. Cordero-Braña: Minimum Hellinger Distance Estimation for Finite Mixture Models. En: *Journal of the American Statistical Association* 91 (1996), Nr. 436, p. 1716–1723. – ISSN 01621459

[3] Agha, M. ; Ibrahim, M. T.: Algorithm AS 203: Maximum Likelihood Estimation of Mixtures of Distributions. En: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 33 (1984), Nr. 3, p. 327–332

[4] Benaglia, Tatiana ; Chauveau, Didier ; Hunter, David R. ; Young, Derek: mixtools: An R Package for Analyzing Finite Mixture Models. En: *Journal of Statistical Software* 32 (2009), Nr. 6, p. 1–29

[5] Beran, Rudolf: Minimum Hellinger Distance Estimates for Parametric Models. En: *The Annals of Statistics* 5 (1977), Nr. 3, p. 445–463. – ISSN 00905364

[6] Crawford, Sybil L.: An Application of the Laplace Method to Finite Mixture Distributions. En: *Journal of the American Statistical Association* 89 (1994), Nr. 425, p. pp. 259–267. – ISSN 01621459

[7] Czarn, A. ; MacNish, C. ; Vijayan, K. ; Turlach, B. ; Gupta, R.: Statistical exploratory analysis of genetic algorithms. En: *IEEE Transactions on evolutionary computation* 8 (2004), Nr. 14, p. 405–421

[8] Dacunha-Castelle, D. ; Gassiat, E.: The Estimation of the Order of a Mixture Model. En: *Bernoulli* 3 (1997), Nr. 3, p. 279–299

[9] Denning, P.J.: The science of computing: Genetic algorithms. En: *American Scientist* 80 (1992), Nr. 1, p. 12–14

[10] Dey, D.K.: Estimation of Scale Parameters in Mixture Distributions. En: *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* 18 (1990), Nr. 2, p. 171–178

[11] E. Susko, J. D. K. ; Chen., J.: Constrained Nonparametric Maximum-Likelihood Estimation for Mixture Models. En: *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* 26 (1998), Nr. 4, p. 601–617

[12] Estrada, J. ; Camacho, J.A ; Restrepo, M.T. ; Parra, C.M.: Parámetros antropométricos de la población laboral colombiana 1995 (acopla95). En: *Rev. Fac. Nac. Salud Pública* 15 (1988), Nr. 2, p. 112–139

[13] Fouskakis, D. ; Draper, D.: Stochastic optimization: a review. En: *International Statistical Review / Revue Internationale de Statistique* 70, Nr. 3

[14] Gallegos, M. ; Ritter, G.: Trimmed ML Estimation of Contaminated Mixtures. En: *Sankhyā: The Indian Journal of Statistics, Series A* 71 (2009), Nr. 2, p. 164–220

[15] Glover, Fred: Tabu Search—Part I. En: *ORSA Journal on Computing* 1 (1989), Nr. 3, p. 190–206

[16] Harrington, Justin: *lga: Tools for linear grouping analysis (LGA)*, 2012. – R package version 1.1-1

[17] Haupt, R.L. ; Haupt., S.E.: *Practical Genetic Algorithms*. Wiley, 2004

[18] Jingjing WU, Rohana J. K.: On minimum Hellinger distance estimation. En: *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* 37 (2009), Nr. 4, p. 514–533. – ISSN 03195724

[19] K. Do, P. M. ; Tang., F.: A Bayesian Mixture Model for Differential Gene Expression. En: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 54 (2005), Nr. 3, p. 627–624

[20] Lahoz-Beltra, R. ; Perales-Gravan., C.: A survey of nonparametric tests for the statistical analysis of evolutionary computational experiments. En: *International Journal Information Theories and Application* 17 (2010), Nr. 1, p. 49–61

[21] McCrea, Rachel S. ; Morgan, Byron J. T. ; Cole, Diana J.: Age-dependent mixture models for recovery data on animals marked at unknown age. En: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 62 (2013), Nr. 1, p. pp. 101–113. – ISSN 00359254

[22] McLachlan, G.J. ; Basford., K.E.: *Mixture models: inference and applications to clustering*. Marcel Dekker, 1988

[23] N. Metropolis, M.N. Rosenbluth A.H. T. ; Teller., E.: Equation of state calculation by fast computing machines. En: *Journal of Chemical Physics* 21 (1953), Nr. 6, p. 1087–1091

[24] Nemec, James ; NEMEC, Amanda F. L.: Mixture models for studying stellar populations. I. Univariate mixture models, parameter estimation, and the number of discrete population components. En: *Publications of the Astronomical Society of the Pacific* 103 (1991), Nr. 659, p. pp. 95–121. – ISSN 00046280

[25] Quintana, Fernando A. ; Newton, Michael A.: Computational Aspects of Nonparametric Bayesian Analysis with Applications to the Modeling of Multiple Binary Sequences. En: *Journal of Computational and Graphical Statistics* 9 (2000), Nr. 4, p. pp. 711–737. – ISSN 10618600

[26] R Core Team: *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2014

[27] R. Tibshirani, G. W. ; Hastie, T.: Estimate the number of clusters in a data set via the gap statistic. En: *Journal of Royal Statistical Society B* 63, Nr. 2

[28] Reschenhofer, E.: The Bimodality Principle. En: *Journal of Statistics Education* 9 (2001), Nr. 1

[29] Reynolds, J.H. ; Templin., W.D.: Comparing Mixture Estimates by Parametric Bootstrapping Likelihood Ratios. En: *Journal of Agricultural, Biological, and Environmental Statistics* 9 (2004), Nr. 1, p. 54–74

[30] S. Chatterjee, M. L. ; Lynch, L.A.: Genetic algorithms and their statistical application: an introduction. En: *Computational Statics and Data Analysis* 22 (1996), Nr. 6, p. 219–234

[31] Santner, T. ; Williams, B. ; Notz, W.: *The Design and Analysis of Computer Experiments*. Springer, 2003. – ISBN 0387954201

[32] Scrucca., L.: GA: A Package for Genetic Algorithms in R. En: *Journal of Statistical Software* 53 (2013), Nr. 4, p. 1–37

[33] Silver., EA.: An overview of heuristic solution methods. En: *Journal of the Operational Research Society* 55 (2004), p. 936–956

[34] Snee., R.D.: Techniques for the analysis of mixture data. En: *Technometrics* 15 (1973), Nr. 3, p. 517–528

[35] Stephens, Matthew: Dealing with Label Switching in Mixture Models. En: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 62 (2000), Nr. 4, p. pp. 795–809. – ISSN 13697412

[36] Tolvi., J.: Genetic algorithms for outlier detection and variable selection in linear regression models. En: *Soft Computing* 8 (2004), p. 527–533

[37] Venables, W. N. ; Ripley, B. D.: *Modern Applied Statistics with S.* Fourth. New York : Springer, 2002. – ISBN 0-387-95457-0

[38] West., M.: Approximating Posterior Distributions by Mixture. En: *Journal of the Royal Statistical Society. Series B (Methodological)* 55 (1993), Nr. 2, p. 409–422

[39] Willighagen, Egon ; Ballings, Michel: *genalg: R Based Genetic Algorithm*, 2015. – R package version 0.2.0

[40] Zhu, M. ; Chipman., H.: Darwinian evolution in parallel universes: A parallel genetic algorithm for variable selection. En: *Technometrics* 48, Nr. 4