International Journal of Development Mathematics Vol 1 Issue 2, Page No. 206 - 217

INTERNATIONAL JOURNAL OF DEVELOPMENT MATHEMATICS

ISSN: 3026-8656 (Print) | 3026-8699 (Online)

journal homepage: https://ijdm.org.ng/index.php/Journals

On Univariate Gaussian Mixture Model for Diabetic Patients in Yola, Adamawa State

Hassan Ahmed[a,c,*], Ahmed Abdulkadir[a], Kazeem E. Lasisi[a], and Rasheed Bello[b]

[a]*Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi, Bauchi State, Nigeria.*

[b]*Department of Mathematical Sciences, Gombe State University, Gombe, Gombe State, Nigeria.*

[c]*Department of Statistics, Modibbo Adama University, Yola, Adamawa State, Nigeria.*

ARTICLE INFO

Article history:
Received 15 January 2024
Received in revised form 12 May 2024
Accepted 20 May 2024

Keywords:
algorithm, Diabetes, mixture-model, EM, mclust, mixtools

MSC 2020 Subject classification:
62P10, 92B15, 62H30

ABSTRACT

Finite Mixture Model serves as a potent statistical tool for modeling intricate data distributions through the amalgamation of multiple Gaussian components. The aim of this work is to use finite mixture of normal distribution to model data on diabetic patients from Modibbo Adama University Teaching Hospital, Yola, Adamawa State, so as to expose latent sub-groups. The model is first used on synthetic data and subsequently on diabetic patients data. The diabetic data set consists of 553 cases of diabetes mellitus. The variables measured: Age(years), Mass of a patient(kg/meters), glucose level (plasma glucose con- centration, a 2-hour in an oral glucose tolerance test), pressure (Diastolic blood pressure mmHg), insulin (2-hour serum insulin mu U/ml) and class variable (0 or 1) treating 0 as false or negative and 1 treated as true or positive test for diabetes. The synthetic data is obtained from Gaussian distribution. The results showed that there is presence of heterogeneity in the data by showing the presence of 2 sub-groups.

1. Introduction

The significance of finite mixture models in statistical data analysis is clear from the escalating frequency of articles addressing both theoretical and practical aspects of these models in statistical and broader scientific literature. This trend is attributed to the widespread adoption of finite mixtures of distributions as computationally efficient representations for modeling intricate data distributions related to random phenomena. Mixture models have found successful applications in diverse fields such as agriculture, astronomy, bioinformatics, biology, economics, engineering, genetics, imaging, marketing, medicine, neuroscience, psychiatry, psychology, and various other disciplines spanning biological, physical, and social sciences (McLachlan *et al*., 2019). Finite mixture models are often used to study data from a population that is suspected to be composed of a number of homogeneous subpopulations (McLachlan *et al*., 2019) .For example, when a disease has a simple genetic cause, the population may be divided into two or three homogeneous groups. In the initial stage of these investigations, it is important to have a sensitive test for the number k of subpopulations included in the data. The construction of such a test, however, is often more challenging than might be expected. Finite mixture models belong to a class of non-regular models and, as a consequence, many classical asymptotic results do not apply.

Finite mixtures of distributions, in particular normal, have been used extensively as models in a wide variety of important practical situations where data can be viewed is arising from two or more populations mixed in varying proportions (Kayal *et al*., 2023). Recent problems which have been addressed by normal mixture models a spatially constrained skew Student's-t mixture model for brain MR image segmentation and bias field correction (Cheng *et al.*, 2022). In addition to this latter role of assessing the performances of estimators in non-normal situations, normal mixtures have been used of course in the development of robust estimators. For example, (Sugasawa *et al*., 2022), observed that under mixtures when they have light tails, such as normal distribution, the mixture model can be sensitive to outliers. It is in the field of cluster analysis, however, that mixtures of distributions, normal or otherwise, are being increasingly used. Various examples of clustering on the basis of normal mixtures may be found in the literature.

Under the mixture likelihood approach to clustering, it is assumed that the observations on the entities to

* Corresponding author. Tel.: +2348032417537
E-mail address: hassanahmed.official@gmail.com (Hassan, A.)
https://doi.org/10.62054/ijdm/0102.17

be clustered are from a mixture of an initially specified number of populations or groups in various proportions. By adopting some parametric form for the density function in each underlying group, a likelihood can be formed in terms of the mixture density, and unknown parameters estimated by consideration of the likelihood. A probabilistic clustering of the entities can then be obtained in terms of their estimated posterior probabilities of group membership. An outright assignment of the entities to the groups can be achieved by allocating each entity to the group to which it has the highest estimated posterior probability of belonging. Although the estimates of the parameters, and hence of the posterior probabilities, may not be reliable if the sample size is not large, a satisfactory clustering as given by this outright assignment may still be achieved (Ganesalingam and McLachlan, 1979).

Decomposing a finite mixture of distributions is a very difficult problem, as shown by early researchers. But with the advent of high-speed computers, attention was turned to likelihood estimation of the parameters in a mixture distribution. The first use of this principle for a mixture model has been attributed to (Rao, 1948) who used Fisher's method of scoring for a mixture of two univariate distributions with equal variances. The Expectation-Maximization (EM) algorithm is a broadly applicable approach to the iterative computation of maximum likelihood (ML) estimates, useful in a variety of incomplete data problems, where algorithms such as the Newton-Raphson method may turn out to be more complicated. On each iteration of the EM algorithm, there are two steps-called the Expectation step or the E-step and the Maximization step or the M-step. Because of this, the algorithm is called the EM algorithm in their fundamental paper (Dempster *et al*,1977). (We hereafter refer to this paper as the DLR paper or simply DLR.) The situations where the EM algorithm is profitably applied can be described as incomplete-data problems, where ML estimation is made difficult by the absence of some part of data in a more familiar and simpler data structure. The EM algorithm is closely related to the ad hoc approach to estimation with missing data, where the parameters are estimated after filling in initial values for the missing data. The latter are then updated by their predicted values using these initial parameter estimates. The parameters are then re-estimated, and so on, proceeding iteratively until convergence. This idea behind the EM algorithm being intuitive and natural, algorithms like the EM have been formulated and applied in a variety of problems even before the DLR paper. But it was in the seminal DLR paper that the ideas were synthesized, a general formulation of the EM algorithm established, its properties investigated, and a host of traditional and non-traditional applications indicated.

Finite mixtures of distributions have been used extensively as models in the field of medicine. Recent works include (Alharithi *et al*., 2021) introduced a novel approach merging discriminative and generative methodologies, termed the Shifted-Scaled Dirichlet Mixture Model (SSDMM) coupled with Support Vector Machines (SVMs), tailored to address complex issues in medical data analysis, particularly in categorization and recognition tasks. The goal of the paper is to accurately capture the intrinsic characteristics of biomedical images by leveraging the strengths of both generative and discriminative models. To realize this objective, a proposal was made and methodology that in- volves several key steps. Firstly, the authors employed robust local descriptor extraction techniques. Subsequently, the Expectation-Maximization (EM) algorithm was used to learn the parameters of the SSDMM, which serves as our generative model. Finally, the authors generated new data- driven SVM kernels from the SSDMM, thus bridging the gap between generative and discriminative approaches. The proposed framework is demonstrated through the resolution of two challenging medical image analysis tasks: the categorization of retinal im- ages into normal or diabetic cases and the identification of lung diseases in chest X-ray (CXR) images. The experimental results highlight the superiority of the hybrid approach over alternative methodologies, showcasing its efficacy in tackling these demanding medical data analysis problems.

Another recent work is of (Aiello *et al*., 2022) where a probabilistic framework for modeling the eating patterns of individuals with type 1 diabetes, validated using extensive datasets gathered from experiments conducted in free-living conditions at clinical centers in Padova and Amsterdam. Meals are conceptualized as discrete-time marked point processes, with random meal occurrences linked to a Markov Chain representing fasting periods. Transition probabilities within the Markov Chain are influenced by factors such as time of

day and carbohydrate in- take from preceding meals. The random marks associated with each meal signify carbohydrate intake, with their distribution potentially influenced by factors such as time of day, fasting duration, and carbohydrate intake from the previous meal.

Logistic and Gaussian Mixture models were employed to derive distinct models from the Padova and Amsterdam datasets, respectively. To assess the validity of the approach, simulation on synthetic datasets using the pro- posed model was carried out, revealing statistical properties consistent with those observed in the experimental datasets. The development of a probabilistic model for eating behavior holds promise for various applications, including the optimization and personalization of automated insulin dosing systems, decision support tools for insulin dosing, and management of alerts for missed meal announcements. The aim of this paper is to model, using Gaussian Mixture, data on diabetic patients from Modibbo Adama University Teaching Hospital, Yola, Adamawa State with the intention of exposing latent sub-groups in the data. The approach is first tested on synthetic data simulated from a probabilistic distribution.

2. Methods

2.1 Learning finite mixture models

A random variable $X = X_1, X_2, \ldots, X_d$ follows a k-component finite mixture distribution, if its probability density function can be written as:

$$\mathrm{p}(x/\theta) = \sum_{m=1}^{\theta} \pi_m \, \mathrm{p}(x/\theta_m) \tag{1}$$

where π_m is the prior probability of the m^{th} component and satisfies:

$$\pi_m \geq 0, \sum_{m=1}^{k} \pi_m = 1 \tag{2}$$

Where θ_m is the parameter of the m^{th} density model and

$$\theta = \{(\pi_m, \theta_m), m = 1, \ldots, k\} \tag{3}$$

is the parameter of the mixture models.

2.2 Derivations of the EM algorithm.

Suppose $X \sim N(\mu,\sigma)$, then the joint distribution of the Gaussian mixture model based on the latent variable Z is given by:

$$\mathrm{p}(Z, X) = \mathrm{p}(Z).\,\mathrm{p}(X|Z) = \mathrm{cat}(\underline{\pi})\mathrm{N}(\mu_z, \sigma_z) \tag{4}$$

$$= \left(\prod_{d=0}^{D-1} \pi_d^{I(d=z)}\right).\left[\frac{1}{\sigma_{z\sqrt{2\pi}}} \exp\left\{\frac{-1}{2\sigma_z^2}(X-\mu_z)^2\right\}\right] \tag{5}$$

2.2.1 E-STEP

The E-step involves obtaining the posterior probabilities, that is, by applying Bayes's rule

$$p\left(Z = z \middle| X = x^{[i]}; \theta^{[k]}\right) \tag{6}$$

$$\frac{p\left(X = x^{[i]} \middle| Z = z; \theta^{[k]}\right) p\left(Z = z; \theta^{[k]}\right)}{p\left(X = x^{[i]}; \theta^{[k]}\right)} \tag{7}$$

$$\sim p\left(X = x^{[i]} \middle| Z = z; \theta^{[k]}\right) \tag{8}$$

$$= \widetilde{\rho_z^{[i]}} \tag{9}$$

$$= \pi_z . \frac{1}{\sigma_{z\sqrt{2\pi}}} \exp\left\{\frac{-1}{2\sigma_z^2}\left(x^{[i]} - \mu_z\right)^2\right\} \tag{10}$$

To obtain the normalized responsibilities, we apply the row normalization,

$$\rho_z^{[i]} = \frac{\widetilde{\rho_z^{[i]}}}{\sum_{d=0}^{D-1} \rho_d^{[i]}} \tag{11}$$

2.2.2 M-STEP

The M-step involves the following

$$Q\left(D; \theta^{[i]}, \theta\right) =$$

$$\sum_{i=0}^{N-1} \sum_{c=0}^{D-1} \rho^{[i]} \left(\sum_{d=0}^{D-1} I(d = c) \log(\pi_d) + \log\left(\frac{1}{\sigma_{c\sqrt{2\pi}}}\right) - \frac{1}{2\sigma_c^2}\left(X^{[i]} - \mu_c\right)^2 \right)$$

$$= \sum_{i=0}^{N-1} \sum_{c=0}^{D-1} \rho^{[i]} \left(\sum_{d=0}^{D-1} I(d = c) \log(\pi_d) - \log(\sigma_c) - \frac{1}{2}\log(2\pi) - \frac{1}{2\sigma_c^2}\left(X^{[i]} - \mu_c\right)^2 \right) \tag{12}$$

Next, we build a Lagrangian to get the unconstrained optimization

$$\widehat{\mathrm{Q}}\left(\mathrm{D};\ \theta^{[\mathrm{k}]}, \theta, \lambda\right) = \hat{Q}\left(D; \theta^{[k]}, \theta\right) + \lambda g(\pi) = \sum_{i=0}^{N-1} \sum_{c=0}^{D-1} \rho^{[i]} \left(\sum_{d=0}^{D-1} I(d = c) \log(\pi_d) + \log\left(\frac{1}{\sigma_{c\sqrt{2\pi}}}\right) - \frac{1}{2\sigma_c^2}\left(X^{[i]} - \mu_c\right)^2 \right)$$

$$= \sum_{i=0}^{N-1} \sum_{c=0}^{D-1} \rho^{[i]} \left(\sum_{d=0}^{D-1} I(d = c) \log(\pi_d) - \log(\sigma_c) - \frac{1}{2\sigma_c^2}\left(X^{[i]} - \mu_c\right)^2 \right) + \lambda\left(\lambda - \sum_{d=0}^{D-1} \pi_c\right) \tag{13}$$

1.

$$\frac{\delta \hat{Q}}{\delta \pi_e} =$$

$$\sum_{i=0}^{N-1} \sum_{c=0}^{D-1} \left(\rho_c^{[i]} I(e = c) \frac{1}{\pi_e}\right) - \lambda = 0$$

$$= \frac{1}{\pi_e} \sum_{i=0}^{N-1} \sum_{c=0}^{D-1} \left(\rho_c^{[i]} I(e = c)\right) - \lambda = 0$$

$$= \frac{1}{\pi_e}\sum_{i=0}^{N-1} \rho_e^{[i]} - \lambda$$

$$set \quad \gamma_e = \sum_{i=0}^{N-1} \rho_e^{[i]}$$

$$\therefore \lambda = \frac{\gamma_e}{\pi_e} \tag{14}$$

$$\frac{\delta \hat{Q}}{\delta \lambda} = \lambda - \sum_{d=0}^{D-1} \pi_d = 0$$

$$\lambda = \sum_{d=0}^{D-1} \pi_d$$

$$\sum_{d=0}^{D-1} \pi_d = 1$$

$$\therefore \lambda = 1 \tag{15}$$

Note

$$\lambda = \lambda$$

$$\lambda . 1 = \lambda . \sum_{d=0}^{D-1} \pi_d$$

$$= \sum_{d=0}^{D-1} \lambda \pi_d$$

$$= \sum_{d=0}^{D-1} \frac{\gamma_d}{N} . \pi_d$$

$$= \sum_{d=0}^{D-1} \pi_d \frac{\gamma_d}{N} = N$$

$$\therefore \lambda = \sum_{d=0}^{D-1} \frac{\gamma_e}{N} \pi_e$$

$$\pi_e = \frac{\gamma_e}{N} \tag{16}$$

2. $$\frac{\delta \hat{Q}}{\delta \mu_e} = \sum_{i=0}^{N-1} \rho_e^{[i]} \left(\frac{-2(X^{[i]} - \mu_e)}{2\sigma_e^2} \right)(-1) = 0$$

$$= \sum_{i=0}^{N-1} \rho_e^{[i]} \left(X^{[i]} - \mu_e \right) = 0$$

$$= \sum_{i=0}^{N-1} \rho_e^{[i]} X^{[i]} - \sum_{i=0}^{N-1} \rho_e^{[i]} \mu_e$$

$$= \sum_{i=0}^{N-1} \rho_e^{[i]} X^{[i]} - \gamma_e \mu_e = 0$$

$$\mu_e = \frac{1}{\gamma_e} \sum_{i=0}^{N-1} \rho_e^{[i]} X^{[i]} \tag{17}$$

3. $$\frac{\delta \hat{Q}}{\delta \sigma_e} = \sum_{i=0}^{N-1} \rho_e^{[i]} \left(\frac{-1}{\sigma_e} + \frac{1}{\sigma_e^3} \left(X^{[i]} - \mu_e \right)^2 \right) = 0$$

$$= \sum_{i=0}^{N-1} \rho_e^{[i]} \left(-\sigma_e^2 + \left(X^{[i]} - \mu_e \right)^2 \right) = 0$$

$$= \sum_{i=0}^{N-1} \rho_e^{[i]} \left(X^{[i]} - \mu_e \right)^2 - \sum_{i=0}^{N-1} \rho_e^{[i]} \sigma_e^2$$

$$\therefore \sigma_e = \sqrt{\frac{1}{\gamma_e} \sum_{i=0}^{N-1} \rho_e^{[i]} (X^{[i]} - \mu_e)^2} \tag{18}$$

2.2.3 Summary of EM for the GMM

Expectation-Step (E-Step)

1. Calculate unnormalized responsibilities:

$$\widetilde{\rho_c^{[i]}} = \pi_c . \frac{1}{\sigma_{c\sqrt{2\pi}}} \exp\left\{\frac{-1}{2\sigma_c^2}\left(x^{[i]} - \mu_c\right)^2\right\} \tag{19}$$

2. The normalized responsibilities:

$$\rho_c^{[i]} = \frac{\widetilde{\rho_c^{[i]}}}{\sum_{d=0}^{D-1} \rho_d^{[i]}} \tag{20}$$

4. Calculate class responsibilities

$$\gamma_c = \sum_{i=0}^{N-1} \rho_c^{[i]} \tag{21}$$

Maximization Step

1.

$$\pi_c = \frac{\gamma_c}{N} \tag{22}$$

2. Update the means

$$\mu_c = \frac{1}{\gamma_c} \sum_{d=0}^{D-1} \rho_d^{[i]} \tag{23}$$

3. Update the sigma

$$\sigma_c = \sqrt{\frac{1}{\gamma_c} \sum_{i=0}^{N-1} \rho_c^{[i]} (X^{[i]} - \mu_c)^2} \tag{24}$$

2.3 Data set and Implementation in R

The used dataset contains simulation data for 1000 records. It is sampled from a 2-component Gaussian Mixture Model. The parameters are as follows:

$\mu_1 = -3$, $\mu_2 = 3$
$\sigma_1 = \sigma_2 = 1$
The dataset generated is in appendix 1. The data are simulated with a grouping variable that splits the sample into two classes of 500 subjects with differing mean and variances. Figure 1 clearly shows that in the simulated data there is considerable heterogeneity in both components. Note that these data are model-driven and merely used for illustration of the software functionality.

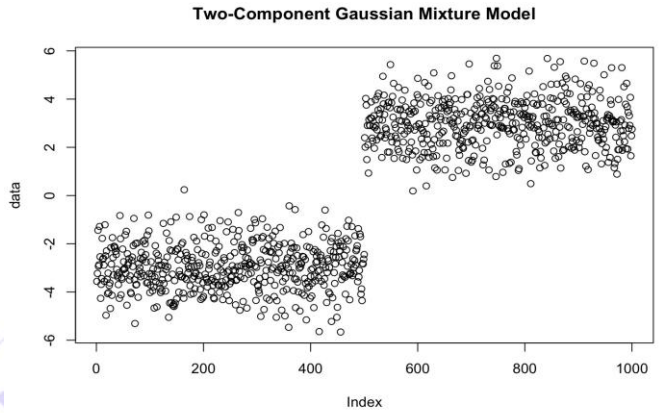

Figure 1: Two-Component Gaussian Mixture Model

2.3.1 Software version

The models are run in R using R-studio. The used versions for this work are R version 4.3.2 (2023-10-31), Platform: x86 64-apple-darwin23.0.0 (64bit), R-studio version: RStudio/2023.09.1+494 For MAC Operating system 14.2.1. The package used is mixtools (Benaglia, Chauveau, Hunter, and Young, 2009) version: 2.0.0 and mclust (Scrucca, Fraley, Murphy, and Raftery, 2023) version 6.0.1. When working analyses, make sure to check the version used and always consult the latest version of the package's reference manual/documentation. Updates and bug-fixes may eventually necessitate alterations to the codes provided below.

3 Results

The 'normalmixEM()' function of the mixtools package can be used to run EM algorithm for a GMM in R. The code is as shown below:

```
# Install and load the mixtools package
install.packages(mixtools) library(mixtools)

# Set a seed for reproducibility
set.seed(123)
lambda1 <- 0.5 mu1 <- -3 sigma1 <- 1
lambda2 <- 0.5 mu2 <- 3 sigma2 <- 1

# Generate some sample data with two components data <- c(rnormmix(500, lambda1, mu1, sigma1), rnorm-
mix(500, lambda2, mu2, sigma2)) typeof(data)
# Fit a two-component Gaussian mixture model gmm _model <- normalmixEM(data, k = 2)
```

We start by opening the package. We used 'library(mixtools)' to load in the and setting the seed that will be used by the random number generator. This allows us to reproduce the results of the procedures with multiple random starts if we rerun the analyses. Next, we generate data from the normal distribution using the 'rnormmix()' function of the 'mixtools' package. The 'rnormmix()' function starts with 'rnormmix(500,)' to generate 500 records. 'lambda1', 'mu1', 'sigma1' are mixing proportion, mean and variance of the first component respectively. 'lambda2', 'mu2', 'sigma2' are the parameters of the second component. Data

generated from the first and second component are saved in a variable which in this case is called 'data'.

The 'typeof()' function allows for checking the type of data. That will enables confirm if the type of data generated is of the required type. The 'normalmixEM()' function is used to implement the EM algorithm as per DLR(1979).

Alternatively, we can use the 'mclust' package to fit the Gaussian mixture. It provides functions for parameter estimation via the EM algorithm for normal mixture models with a variety of covariance structures, and functions for simulation from these models. Also included are functions that combine model-based hierarchical clustering, EM for mixture estimation and the Bayesian Information Criterion (BIC) in comprehensive strategies for clustering, density estimation and discriminant analysis. Additional functionalities are available for displaying and visualizing fitted models along with clustering, classification, and density estimation results. After generating the data, we run the code below

```
# Install and load the mixtools package
install.packages(mclust)
library(mclust)
# Set a seed for reproducibility
set.seed(123)
lambda1 <- 0.5
mu1 <- -3
sigma1 <- 1
 lambda2 <- 0.5
 mu2 <- 3
 sigma2 <- 1
# Generate some sample data with two components data <- c(rnormmix(500, lambda1, mu1, sigma1), rnorm-
mix(500, lambda2, mu2, sigma2)) typeof(data)
# Fit a two-component Gaussian mixture model gmm _model <- Mclust(data, k = 2)
```

This is just like the code in 'mixtools' package. But the 'mclust' provides more functions on visualizing the results. For example, the code above for 'mclust' could be re-written as below, for convenience, we skip the data generation part.

```
mod1 <- Mclust(data, x = BIC) summary(mod1, parameters = TRUE) BIC
<- mclustBIC(data) plot(BIC)
summary(BIC) plot(mod1, what = "classification")
```

The 'Mclust' function performs the estimation. The 'summary()' function with parameters set to TRUE shows the results on Table 2. The 'mclustBIC()' fits the data and captures the BIC. Which can then be used for visualization as seen on Figure 2 below. The 'summary()' can once again be used to select the best BIC. Figure 3 is the plot of BIC against classification showing the performance of the selected model.

Table 1: Parameter Estimates of the Simulated data

	Component 1	Component 2
Mixing Probability	0.4994902	0.5005098
Means	-2.968253	2.99443
Variances	0.9833111	0.9833111
Clustering Table	499	501

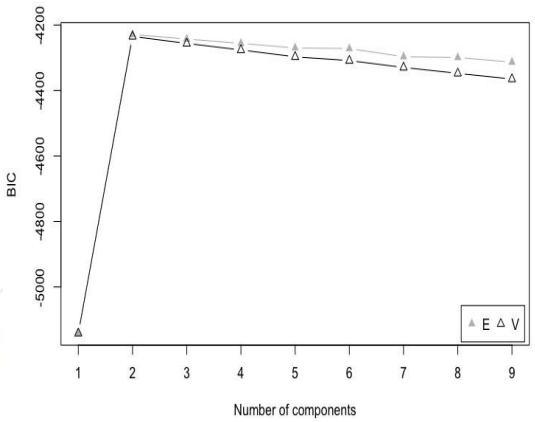

Figure 2: BIC against number of components

Table 2: Log Likelihood

Log-likelihood	n	df	BIC	ICL
-2100.788	1000	4	-4229.208	-4231.018

Table 3: Summary of best model

	E2	V2	E3
BIC	-4229.208	-4234.8944	-4242.92800
BIC diff	0.000	-5.6864	-13.72005

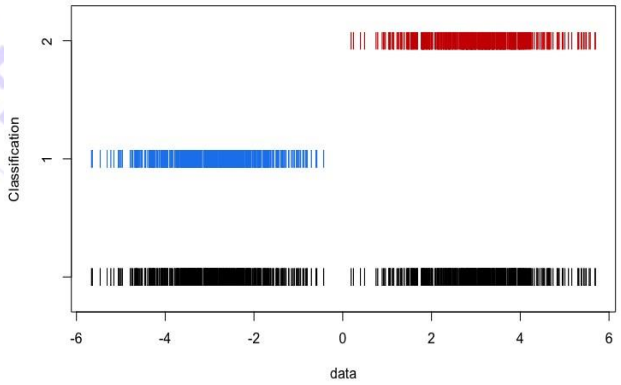

Figure 3: Classification performance

From Table 1, it can be seen that EM algorithm correctly selected the model with 2 components. Component 1 has an estimated mixing proportion of 0.499 while component 2 has 0.500. The algorithm also estimated the parameters of GMM accurately. For Component 1, the mean is -2.96 compared with 3 from the simulated data. The variance is 0.98 which is close to 1 from the simulated data. For component 2, the estimated mean and variance is 2.99 and 0.98 respectively. This can be compared with 3 and 1 from the simulated data. Figure 2 shows the BIC level of the selected model. It is seen clearly that the 2-component model has BIC of -4200. Table 2 confirmed that by displaying the Log-likelihood = -2100.78, BIC = -4229.98 and ICL = -4231.018. Figure 3 is showing the classification performance of the model. The figure is showing how the model classifies each observation from the data into the right class or component. We observe that all observations are correctly classified. This can be observed from Figure 3, the black lines can be traced to either the blue(component 1) or the red(component 2).

3.1 Application to real data set

The data to be used is due to (Ahmed, Mohammed, and Baba, 2021). The data set consists of 553 cases of diabetes mellitus that were collected at Federal Medical Center, Yola. The variables measured: Age(years), Mass of a patient(kg/meters), glucose level (plasma glucose concentration, a 2-hour in an oral glucose tolerance test), pressure (Diastolic blood pressure mmHg), insulin (2-hour serum insulin mu U/ml) and class variable (0 or 1) treating 0 as false or negative and 1 treated as true or positive test for diabetes.
Figure 3 below shows the plot of the variables against the dependent variable class. It clearly shows the presence of mixture in all the variables.

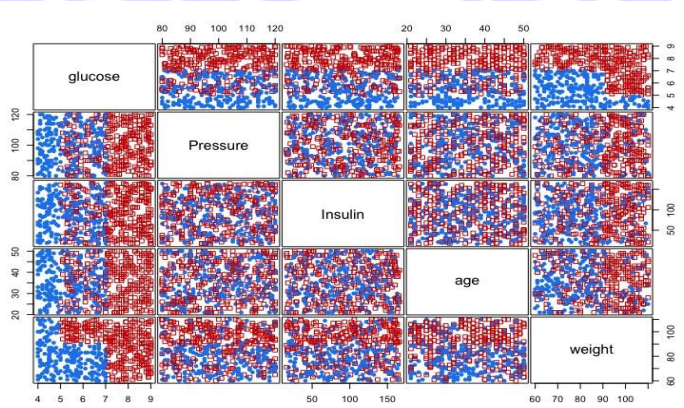

Figure 4: Visualization of data. Dependent variable against all independent variables

The table below shows the results and estimates of the fitted 2 - component mixture model.

Table 4: estimates of 2 component Gaussian mixture model for test data.

	Component 1	Component 2
Mixing Probability:	0.5033172	0.4966828
Means:		
glucose	6.535202	6.510566
Pressure	100.454924	100.278080
Insulin	93.272307	90.721621

age	35.066933	36.297469
weight	96.882812	73.249726
Variances:		
glucose	2.17565	2.17565
Pressure	143.5265	143.5265
Insulin	1918.61	1918.61
age	75.68422	75.68422
weight	64.8247	64.8247
Clustering Table	499	501

The code used in generating the results above is given by:

```
library(mclust) class <- data$class X <- data[,-6] clPairs(X,
class) BIC <- mclustBIC(X) plot(BIC) summary(BIC) mod1 <-
Mclust(X, x = BIC) summary(mod1, parameters = TRUE)
mod2 <- Mclust(data, x = BIC) summary(mod2, parameters =
TRUE)
```

The 'clPairs()' function of the 'mclust()' package is used plot the independent variables: glucose, age, weight, insulin; against the dependent variable class. The plot is shown on figure 3 above.
The functions available in the package can be modified to suit a specific need of researchers.

Similar to the case of the simulated data, the diabetic data shows the presence of two components or sub groups. Each of the component has a mixing proportion of 0.5 and 0.49 respectively. This is indicative of presence of 2 normal distributions mixed within the data. Table 4 shows a summary of the estimates of the variables measured for all patients. It is however worth noting that computational load for the real data set is high. This can be an area for further investigation. The one-component, four-component models could not converge and as a result the algorithm has to be manually terminated. Implementation of a further technique to handle such scenario is the subject of an on-going investigation by the authors of this work.

4. Conclusion

This work used Gaussian mixture model, a type of finite mixture, to model diabetic data. We first tested the methodology on synthetic data-set. Then we modeled the real data set. The analysis showed clearly the presence of 2 sub-groups. Finite mixtures have the potential to provide additional insights on the relationship be- tween these sub groups. It should be more widely considered as a method of investigation in this area. A potential future work is to use mixture of other probabilistic distributions to fit to the data. This will further enhance accuracy of estimation of parameters and determining the number of components in the mixture.

References

Ahmed, H., Mohammed, M. B., and Baba, I. A. (2021). On Comparing Multi-Layer Perceptron and Logistic Regression For Classification Of Diabetic Patients In Federal Medical Center Yola, Adamawa State. *International Journal of Engineering Technonologies and Management research*, 8(6)

Aiello, E. M., Toffanin, C., Magni, L., and De Nicolao, G. (2022). Model- based identification of eating behavioral patterns in populations with type 1 diabetes. *Control Engineering Practice*, 123, 105128.

Alharithi, F., Almulihi, A., Bourouis, S., Alroobaea, R., and Bouguila, N. (2021). Discriminative learning approach based on flexible mixture model for medical data categorization and recognition. *Sensors*, 21(7), 2450.

Benaglia, T., Chauveau, D., Hunter, D. R., and Young, D. (2009). mixtools: An R Package for Analyzing Finite Mixture Models. *Journal of Statistical Software*, 32(6), 1–29. Retrieved from https://www.jstatsoft.org/v32/i06/

Cheng, K. K., Lam, T. H., and Leung, C. C. (2022). Wearing face masks in the community during the COVID-19 pandemic: altruism and solidarity. The Lancet, 399(10336), e39–e40.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society*: series B (methodological), 39(1), 1–22.

Ganesalingam, S., and McLachlan, G. J. (1979). Small sample results for a linear discriminant function estimated from a mixture of nor- mal populations. *Journal of Statistical Computation and Simu- lation*, 9(2), 151–158.

Kayal, S., Bhakta, R., and Balakrishnan, N. (2023). Some results on stochastic comparisons of two finite mixture models with general components. *Stochastic Models*, 39(2), 363–382.

McLachlan, G., Lee, S. X., and Rathnayake, S. I. (2019). Finite mixture models. *Annual review of statistics and its application*, 6, 355– 378.

Rao, C. R. (1948). The utilization of multiple measurements in prob- lems of biological classification. *Journal of the Royal Statistical Society*. Series B (Methodological), 10(2), 159–203.

Scrucca, L., Fraley, C., Murphy, T. B., and Raftery, A. E. (2023). Model-Based Clustering, Classification, and Density Estimation Using mclust in R. *Chapman and Hall/CRC*. Retrieved from https://mclust-org.github.io/book/ doi: 10.1201/9781003277965

Sugasawa, S., Kim, J. K., and Morikawa, K. (2022). Semiparametric imputation using latent sparse conditional Gaussian mixtures for multivariate mixed outcomes. preprint arXiv:2208.07535 .