Lab 1 : Mixture models

Fundamentals of Probabilistic Data Mining
Master 2 MSIAM and MOSIG

Daniel Gabaï
daniel.gabai@grenoble-inp.org
Suraj Ghimire
suraj.ghimire@grenoble-inp.org
Habib Slim
habib.slim@grenoble-inp.org
Sofiane Tanji
sofiane.tanji@grenoble-inp.org

Supervised by **Xavier Alameda-Pineda** and **Thomas Hueber**

Introduction

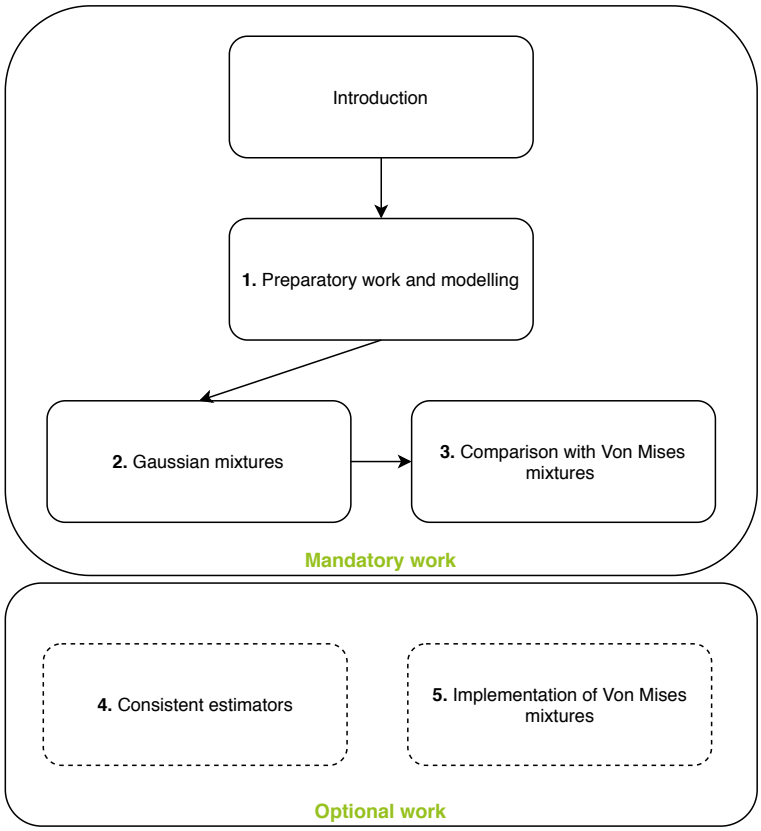
This first lab session allowed us to manipulate mixture models using a small provided dataset. This report aims at answering the questions asked throughout the lab while displaying our understanding of mixture models.

The Unistroke Alphabet is a pattern recognition system for input introduced by Goldberg and Richardson in 1993 and is based on the use of monographs that can be written on PDAs. The letters are simplified in order to increase the recognition rate of the system. The dataset provided contains 50 time-trajectories representing each of the six letters A, E, H, L, O and Q. We will work mainly on the letter A but our work can easily be extended to other letters.

The main issue that this lab work seeks to address is to model the written letters of the dataset using mixture models. This document summarizes our attempt to answer the problem statement..

We also provide a Jupyter notebook alongside this report with all of our code, which follows the same outline as the lab subject.

DIAGRAMMATIC OUTLINE OF THE REPORT



Outline

1 Preparatory work and modelling

1.1 Reestimation formula for GMMs

Let us denote the observations: $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, the latent variables: $\boldsymbol{Z} = \{z_n\}_{n=1}^N$, the parameters: $\boldsymbol{\Theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, the mixture coefficients: $\pi_k = p(z_n = k)$ and the probability: $p(\boldsymbol{x}_n|z_n = k) = \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
Moreover, the posterior probability that $\boldsymbol{x}_n$ belongs to group k is written as: $\eta_{nk} = p(z_n = k|\boldsymbol{x}_n; \boldsymbol{\Theta}^0)$.
The expected complete-data log-likelihood $\mathcal{Q}$ is then defined as:

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^0) &= \mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X};\, \boldsymbol{\Theta}^0)} \log p(\boldsymbol{Z}|\boldsymbol{X}; \boldsymbol{\Theta}) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} p(z_n = k|\boldsymbol{x}_n; \boldsymbol{\Theta}^0) \log p(\boldsymbol{x}_n, z_n|\boldsymbol{\Theta}) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} \eta_{nk} \log\left(\pi_k\, \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)\end{aligned}$$

We are now to find the re-estimation formula in that case. In order to do that, we compute the parameters $\boldsymbol{\mu}_k^*$, $\boldsymbol{\Sigma}_k^*$ and the mixture coefficients π_k^* by optimizing the objective function:

$$\boldsymbol{\Theta}^1 = \arg\max_{\boldsymbol{\Theta}} \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^0)$$

To get $\boldsymbol{\mu}_k^*$, we do:

$$\begin{aligned}\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_k} &= \frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{n=1}^{N}\sum_{j=1}^{K} \eta_{nj} \log\Big(\pi_j\, \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\Big) \\ &= \sum_{n=1}^{N} \eta_{nk} \frac{\partial}{\partial \boldsymbol{\mu}_k} \log\left(\mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)\end{aligned}$$

We have (where d denotes the dimension of the data points $\boldsymbol{x}_n$.):

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\mu}_k} \log\left(\mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) &= \frac{\partial}{\partial \boldsymbol{\mu}_k} - \frac{1}{2}\Big[d \log 2\pi + \log|\boldsymbol{\Sigma}_k| + (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)\Big] \\ &= -\frac{1}{2}\frac{\partial}{\partial \boldsymbol{\mu}_k}\Big[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T {\boldsymbol{\Sigma}_k}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)\Big] \\ &= -{\boldsymbol{\Sigma}_k}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\end{aligned}$$

Using the property $\frac{\partial}{\partial x} x^T A x = (A + A^T)x$, and the fact that ${\boldsymbol{\Sigma}_k}^{-1}$ is symmetric. Plugging this result back into the previous relation, we get:

$$\frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_k} = -\sum_{n=1}^{N} \eta_{nk}\, {\boldsymbol{\Sigma}_k}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)$$

Cancelling and solving for $\boldsymbol{\mu}_k$ then gives us $\boldsymbol{\mu}_k^*$:

$$\Leftrightarrow N{\boldsymbol{\Sigma}_k}^{-1} \sum_{n=1}^{N} \eta_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k^*) = 0$$

$$\Leftrightarrow \sum_{n=1}^{N} \eta_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k^*) = 0$$

$$\Leftrightarrow \boldsymbol{\mu}_k^* = \frac{1}{S_k} \sum_{n=1}^{N} \eta_{nk}\, \boldsymbol{x}_n \quad \square$$

With S_k defined as $\sum_{n=1}^{N} \eta_{nk}$.
For $\boldsymbol{\Sigma}_k^*$, we proceed in a similar way:

$$\begin{aligned}
\frac{\partial}{\partial {\boldsymbol{\Sigma}_k}^{-1}} \log\left(\mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) &= \frac{\partial}{\partial {\boldsymbol{\Sigma}_k}^{-1}} - \frac{1}{2}\left[d \log 2\pi + \log|\boldsymbol{\Sigma}_k| + (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T {\boldsymbol{\Sigma}_k}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right] \\
&= -\frac{1}{2}\frac{\partial}{\partial {\boldsymbol{\Sigma}_k}^{-1}}\left[-\log|{\boldsymbol{\Sigma}_k}^{-1}| + (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T {\boldsymbol{\Sigma}_k}^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right] \\
&= \frac{1}{2}\left(\boldsymbol{\Sigma}_k - (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\right)
\end{aligned}$$

Using properties $|A^{-1}| = \frac{1}{|A|}$, $\frac{\partial \log|X|}{\partial X} = (X^{-1})^T$, $\frac{\partial a^T X a}{\partial X} = aa^T$, and the fact that ${\boldsymbol{\Sigma}_k}^{-1}$ is symmetric. We then have:

$$\frac{\partial \mathcal{Q}}{\partial {\boldsymbol{\Sigma}_k}^{-1}} = \frac{1}{2}\sum_{n=1}^{N} \eta_{nk}\left(\boldsymbol{\Sigma}_k - (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\right)$$

Finally, to obtain $\boldsymbol{\Sigma}_k^*$ we have:

$$\frac{1}{2}\sum_{n=1}^{N} \eta_{nk}\left(\boldsymbol{\Sigma}_k^* - (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\right) = 0$$

$$\Leftrightarrow \boldsymbol{\Sigma}_k^* = \frac{1}{S_k}\sum_{n=1}^{N} \eta_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \quad \square$$

To optimize the mixing coefficients π_k, we need to enforce the constraint that π must be a stochastic vector. We introduce the following Lagrangian for π to solve this problem, with λ being the Lagrange multiplier:

$$\mathcal{L}_\pi = \sum_{n=1}^{N}\sum_{k=1}^{K} \eta_{nk} \log \pi_k + \lambda\left(\sum_{k=1} \pi_k - 1\right)$$

We then have:

$$\frac{\partial \mathcal{L}_\pi}{\partial \pi_k} = \sum_{n=1}^{N} \eta_{nk}\, \frac{1}{\pi_k} + \lambda \tag{1}$$

Cancelling to find ${\pi_k}^*$, we get:

$$\frac{\partial \mathcal{L}_\pi}{\partial \pi_k} = 0 \Leftrightarrow {\pi_k}^* = -\frac{\sum_{n=1}^{N} \eta_{nk}}{\lambda} = -\frac{S_k}{\lambda}$$

Summing both sides over $[\![1, K]\!]$, we obtain:

$$-\frac{\sum_{k=1}^{K} S_k}{\lambda} = 1 \Leftrightarrow \lambda = -N$$

Indeed, we show that $\sum_{k=1}^{K} S_k = N$ in the following way:

$$\begin{aligned}\sum_{k=1}^{K} S_k &= \sum_{k=1}^{K}\sum_{n=1}^{N} \eta_{nk} = \sum_{n=1}^{N}\sum_{k=1}^{K} \eta_{nk} \\ &= \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \mathcal{N}(\boldsymbol{x}_n, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = N\end{aligned}$$

Which is intuitive since S_k measures a (fuzzy) quantity of data points belonging to group k, summing over all S_k should thus give the total number of points.
Plugging back $\lambda = -N$ in relation (1) finally gives us:

$$\sum_{n=1}^{N} \eta_{nk} \frac{1}{{\pi_k}^*} - N = 0 \Leftrightarrow {\pi_k}^* = \frac{S_k}{N} \quad \square$$

In summary, we obtained the following reestimation relations for a Gaussian Mixture Model:

$$\begin{aligned}\pi_k^* &= \frac{1}{N} S_k \\ \boldsymbol{\mu}_k^* &= \frac{1}{S_k} \sum_{n=1}^{N} \eta_{nk}\, \boldsymbol{x}_n \\ \boldsymbol{\Sigma}_k^* &= \frac{1}{S_k} \sum_{n=1}^{N} \eta_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k^*)(\boldsymbol{x}_n - \boldsymbol{\mu}_k^*)^T\end{aligned}$$

1.2 Simulation of a bivariate GMM

We now want to simulate a sample of size 500 of the bivariate Gaussian Mixture Model:

$$\mathcal{M} \sim 0.3\, \mathcal{N}(\boldsymbol{\mu}_1; \boldsymbol{\Sigma}_1) + 0.7\, \mathcal{N}(\boldsymbol{\mu}_2; \boldsymbol{\Sigma}_2)$$

Where:

$$\boldsymbol{\mu}_1 = \begin{pmatrix} -3 \\ 0 \end{pmatrix}, \boldsymbol{\mu}_2 = \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \boldsymbol{\Sigma}_1 = \begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix}, \boldsymbol{\Sigma}_2 = \begin{pmatrix} 5 & 2 \\ 2 & 2 \end{pmatrix}.$$

A 3D plot of the PDF of this mixture model is shown in figure 1.
In order to sample from the mixture distribution $\mathcal{M}$, we can for each point p:

- Sample k from a Bernoulli distribution of support $[\![1, 2]\!]$, with $p(k = 1) = 0.3$ and $p(k = 2) = 0.7$.
- Depending on the value of k, sample p from $\mathcal{N}(\boldsymbol{\mu}_k; \boldsymbol{\Sigma}_k)$.

This can be extended to any number K of components in the mixture by sampling from a categorical distribution of support $[\![1, K]\!]$ and probability mass function such that we have $p(k = i) = \pi_i$.
Using the `numpy.random.multivariate_normal` function, we simulate a sample of size 500 using the method above. A 2D plot of this sample is shown in figure 2, with each color representing one Gaussian component. The code used for both figures is provided in the Jupyter notebook attached.

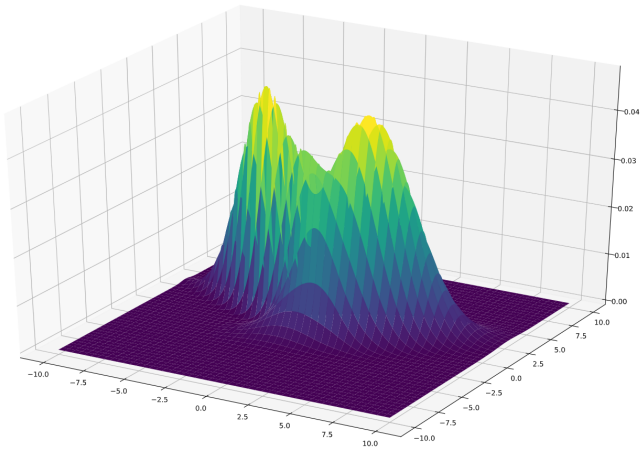


Figure 1: Plotting the PDF of the mixture distribution $\mathcal{M}$

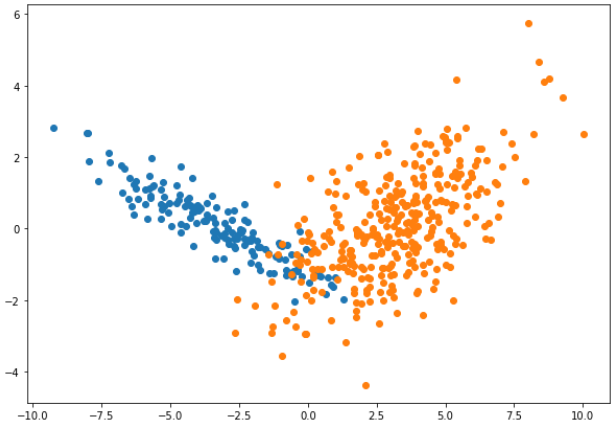


Figure 2: Sampling 500 points from the above GMM

We can clearly see that that the shape of the PDF in figure 1 matches with the distribution of the sampled points in figure 2, as expected.

1.3 Exploration of the dataset

We were given a dataset containing 50 time trajectories per letter for six letters: A, E, H, L, O and Q. Each file represents a list of coordinates which indicates the position of a pointer at a time t.

```
1  −1.000000      139.000000
2  149.000000     116.000000
3  144.000000     107.000000
4  135.000000     90.000000
```

Listing 1: Inside of the file Unistroke/A32.txt

We first plot all the coordinates of every time-trajectory for the letter A. Notice that the

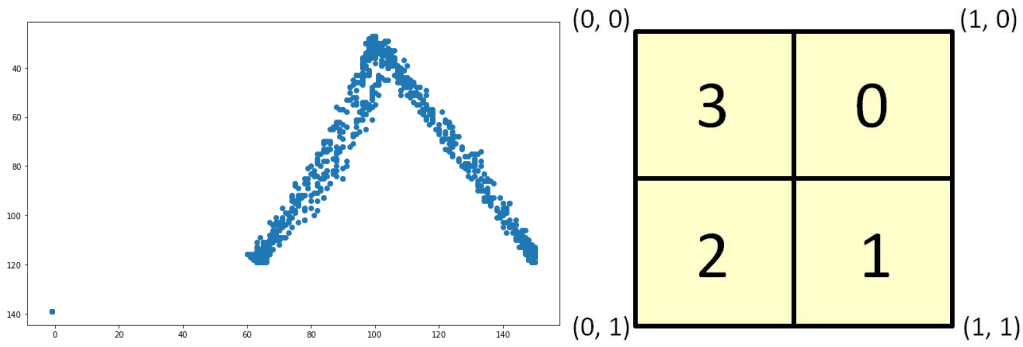


Figure 3: Plot of the Unistroke letter A (left), Unistroke bounding box (right)

figure above has its y-axis inverted. This is because of the Unistroke format for which the y-axis is itself inverted compared to usual, as seen in the bounding box figure above. However, we choose to work with the directions instead of the positions.

Rather than working on the tuples $(x[t], y[t])$ and $(x[t+1], y[t+1])$, we will work on the associated vector $(x[t+1] - x[t], y[t+1] - y[t])$ normalized. The figure below is a plot of the aggregated direction vectors for the letter A.

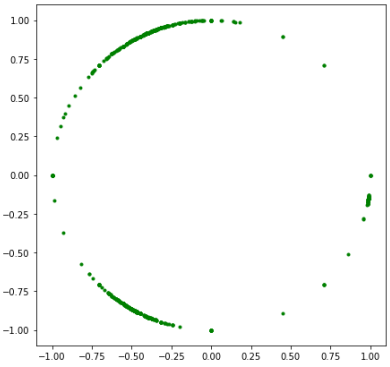


Figure 4: Plot of the directions for the Unistroke letter A

From now on, when referring to the dataset, we refer to the modified dataset with the normalized direction vectors rather than the positions.

1.4 A 2-components GMM for the letter A

In figure 4, we notice two very distinct groups (with a high density of points) and a small one (around $(0.9, -0.1)$). It thus seems reasonable to use a 2-components GMM to model the letter A, each component fitting one of the two strokes which constitute the letter A (drawn as a Λ on a PDA).

2 Gaussian mixtures

2.1 Estimating a GMM

Using the `mixture` module from sklearn, we fit the normalized directions with a two-component bivariate GMM, and we obtain the following parameters (rounded to 10^{-3}):

$$\mu_1 = \begin{pmatrix} -0.388 \\ 0.865 \end{pmatrix},\ \mu_2 = \begin{pmatrix} -0.268 \\ -0.781 \end{pmatrix},\ \Sigma_1 = \begin{pmatrix} 0.063 & 0.036 \\ 0.036 & 0.038 \end{pmatrix},\ \Sigma_2 = \begin{pmatrix} 0.257 & 0.105 \\ 0.105 & 0.061 \end{pmatrix},$$
$$\pi_1 = 0.503,\ \pi_2 = 0.497$$

2.2 Labelling the data

Data points are then labelled by taking the arg max of the responsibilities η_{nk} (essentially what the `predict` function does):

$$\widehat{k}_x = \underset{k \in [\![1,K]\!]}{\arg\max}\ p(z_n = k | \boldsymbol{x}; \boldsymbol{\Theta})$$

In figure 5 is a representation of the data labelled in two classes (green and blue), alongside the contour levels (in log scale) of the PDF of the estimated mixture of parameters given in 2.1.

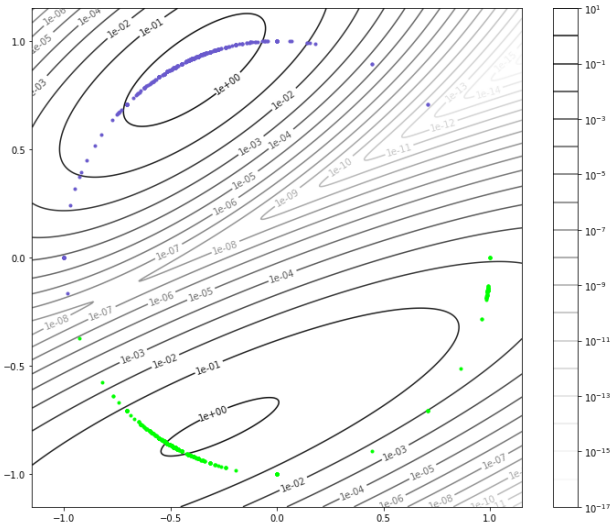


Figure 5: Predictions of the GMM on the data, with contour levels of the GMM PDF

We can see that the vectors oriented towards the $[-1, -1]$ quadrant are not very well adjusted by the GMM, because it also fits the smaller cluster of points around coordinates $[1, 0]$. In order to account for this problem, a third component can be added to the previous mixture. The result can be seen in figure 6.
With a 3-components bivariate GMM, each cluster of normalized vectors is now properly adjusted by a Gaussian component. As seen as in figure 17 of the appendix, adding an additional component does not seem to further improve the fit.

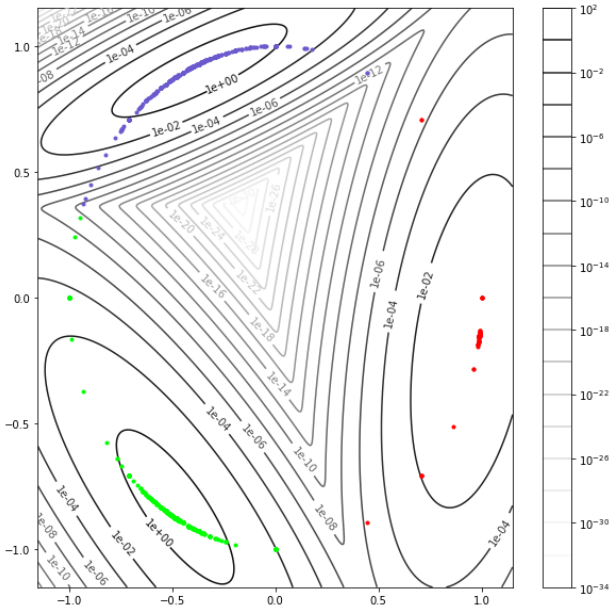


Figure 6: Predictions and PDF contour levels for a GMM with $K = 3$

2.3 Validating the bivariate GMM

To validate the bivariate model, we compute the marginal histograms in x and y of the normalized directions and fit 2-components **univariate** gaussian mixtures for both histograms. We obtain the two figures below:

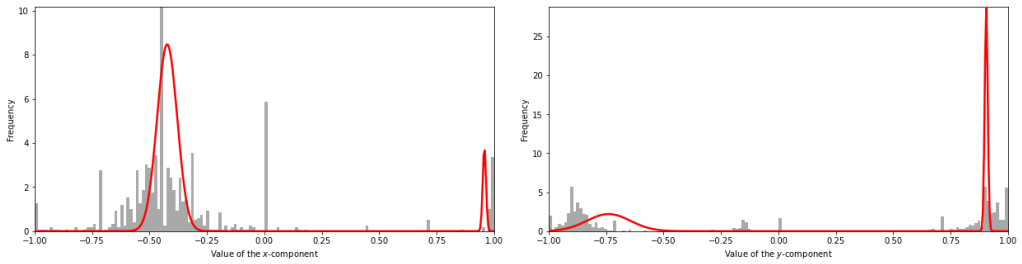


Figure 7: Marginal histograms in x (left) and y (right) with PDF of estimated univariate mixtures of Gaussians

We then separate each cluster and plot separate histograms for x and y, as shown in figure 8 on the following page.

2.4 Comments about the bivariate assumption

We can see that this univariate fit seems suboptimal and fails to capture groups of interest. In figure 8, this is more visible: we can see that the Gaussian approximation of data groups fails to properly fit the marginal distributions, and increasing the number of components per dimension from 2 to a higher value would be unsufficient.

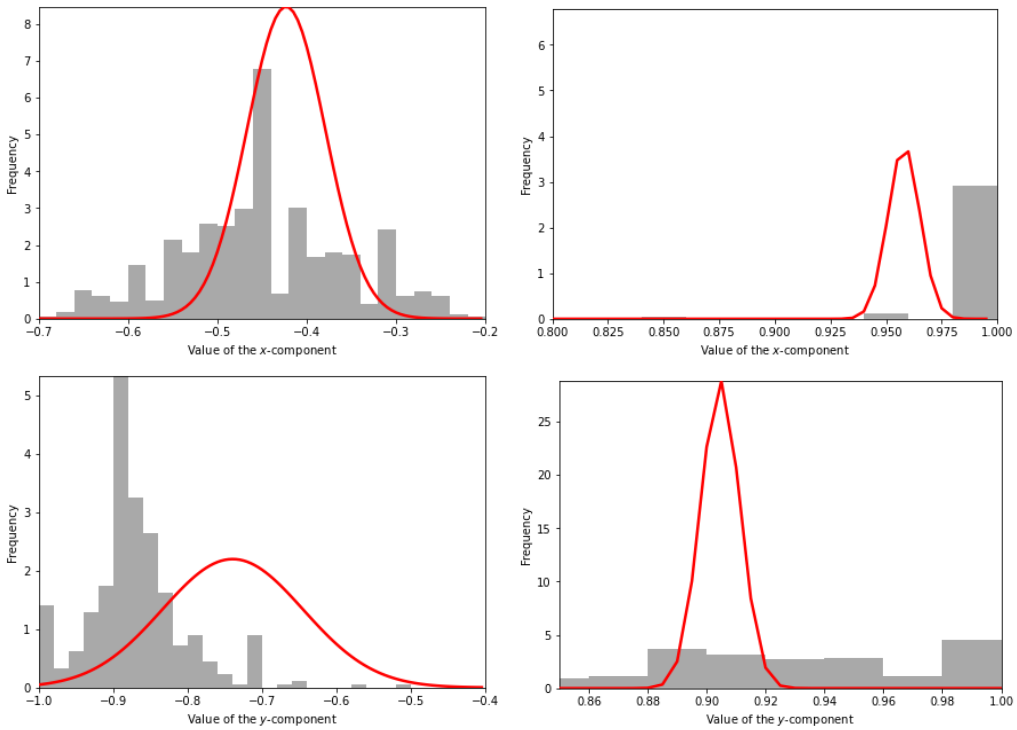


Figure 8: Isolating the two major cluster of points in the histograms

2.5 Plotting posterior probabilities

The posterior probabilities $\Pr(\boldsymbol{Z}_i = 1|\boldsymbol{X}_i)$ are given in figure 9 below, in logarithmic scale. This uses the previously estimated bivariate 2-component GMM of section 2.1.

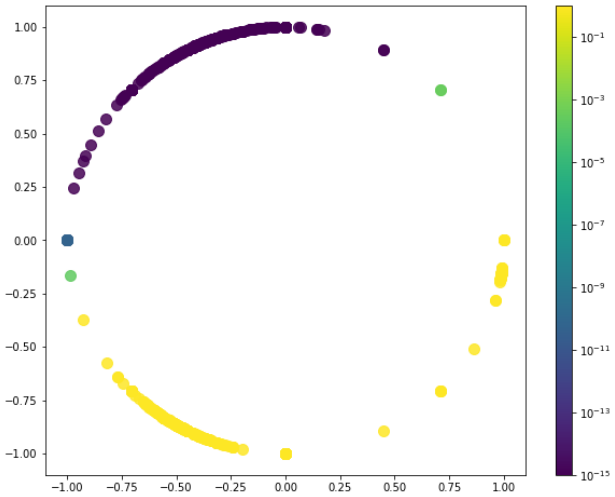


Figure 9: Posterior probabilities η_{nk} for $k = 1$, in logarithmic scale

This plot gives us the probability for each point to belong in group 1 or 2: this classifies all points into both categories. This plot can thus be interpreted as a soft clustering of the dataset, for two clusters ($K = 2$).

3 Comparison with Von Mises mixtures

3.1 Transformation into angular data

Figure 10 below represents the histogram of the dataset transformed into angular data.

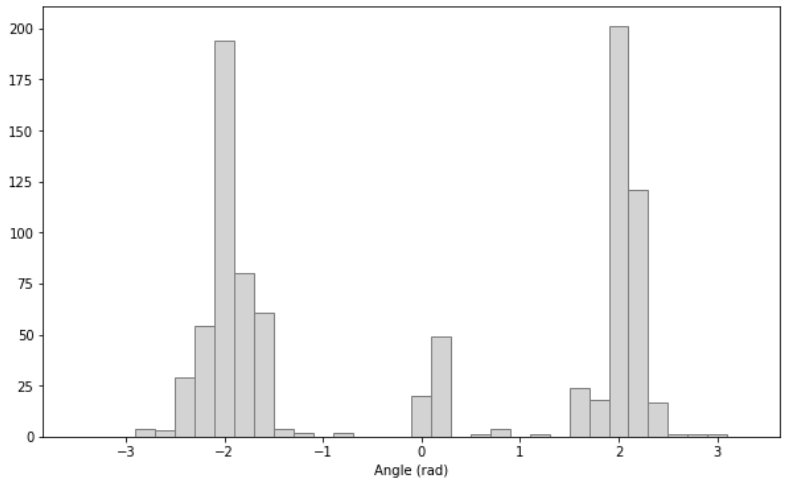


Figure 10: Histogram of angular data

We notice that the angle value is sufficient to clearly distinguish the three groups that we are interested in capturing, here approximately centered around rads values $\{-2, 0, 2\}$.

3.2 Mixtures of Von Mises distributions

For parameters $\mu \in \mathbb{R}, \kappa \in \mathbb{R}_+^\star$, the Von Mises probability density function is given by:

$$f(x|\mu, \kappa) = \frac{e^{\kappa \cdot \cos(x-\mu)}}{2\pi I_0(\kappa)}$$

where $I_0(\kappa)$ is the modified Bessel function of order 0 evaluated at κ as defined below:

$$I_0(\kappa) = \frac{1}{\pi} \int_0^\pi e^{\kappa \cdot \cos(x)}\, dx$$

The Von Mises distribution is the circular analogue of the normal distribution: it should be logical that its fits angular data better than the Gaussian model.
The histogram in figure 10 confirms this point: angles seems to follow a mixture of Von Mises distributions.

3.3 Von Mises EM algortihm

Let us denote the observations: $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$, the latent variables: $\boldsymbol{Z} = \{z_k\}_{k=1}^K$, the parameters: $\boldsymbol{\Theta} = \{\pi_k, \mu_k, \kappa_k\}_{k=1}^K$, the mixture coefficients: $\pi_k = p(z_n = k)$ and the probability: $p(\boldsymbol{x}_n|z_n = k) = \frac{e^{\kappa_k \cos(x_n - \mu_k)}}{2\pi I_0(\kappa_k)}$.

E-step:
The posterior probability that $\boldsymbol{x}_n$ belongs to group k is written as: $\eta_{nk} = p(z_n = k|\boldsymbol{x}_n; \boldsymbol{\Theta}^0)$.

$$\eta_{nk} = \frac{\pi_k f(x_n|\mu_k, \kappa_k)}{\sum_{l=1}^{K} \pi_l f(x_n|\mu_l, \kappa_l)}$$

The expected complete-data log-likelihood $\mathcal{Q}$ is:

$$\mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^0) = \sum_{n=1}^{N} \sum_{k=1}^{K} \eta_{nk} \log \left(\pi_k \frac{e^{\kappa_k \cos(x_n - \mu_k)}}{2\pi I_0(\kappa_k)} \right)$$

M-step:

$$\boldsymbol{\Theta}^1 = \arg\max_{\boldsymbol{\Theta}} \mathcal{Q}(\boldsymbol{\Theta}, \boldsymbol{\Theta}^0)$$

To get $\boldsymbol{\mu}_k^*$, we compute:

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial \boldsymbol{\mu}_k} &= \sum_{n=1}^{N} \eta_{nk} \frac{\partial}{\partial \boldsymbol{\mu}_k} \log \left(\frac{e^{\kappa_k \cos(x_n - \mu_k)}}{2\pi I_0(\kappa_k)} \right) \\ &= \sum_{n=1}^{N} \eta_{nk} \ \kappa_k \sin(x_n - \mu_k) \end{aligned}$$

So, we have $\boldsymbol{\mu}_k^*$ such that:

$$\sum_{n=1}^{N} \eta_{nk} \ \sin(x_n - \mu_k^*) = 0$$

$$\sum_{n=1}^{N} \eta_{nk} (\sin(x_n)\cos(\mu_k^*) - \cos(x_n)\sin(\mu_k^*)) = 0$$

So, we have:

$$tan(\mu_k^*) = \frac{\sum_{n=1}^{N} \eta_{nk} \sin(x_n)}{\sum_{n=1}^{N} \eta_{nk} \cos(x_n)}$$

$$\mu_k^* = arctan \left(\frac{\sum_{n=1}^{N} \eta_{nk} \sin(x_n)}{\sum_{n=1}^{N} \eta_{nk} \cos(x_n)} \right) \quad \square$$

To get κ_k^*, we do:

$$\begin{aligned} \frac{\partial \mathcal{Q}}{\partial \kappa_k} &= \sum_{n=1}^{N} \eta_{nk} \frac{\partial}{\partial \kappa_k} \log \left(\frac{e^{\kappa_k \cos(x_n - \mu_k)}}{2\pi I_0(\kappa_k)} \right) \\ &= \sum_{n=1}^{N} \eta_{nk} \left(\cos(x_n - \mu_k) - \frac{I_1(\kappa_k)}{I_0(\kappa_k)} \right) \end{aligned}$$

With $I_1(\kappa) = \frac{1}{\pi} \int_0^{\pi} \cos(x) e^{\kappa \cos(x)} \, dx$
So, we have κ_k^* such that:

$$\sum_{n=1}^{N} \eta_{nk} \left(\cos(x_n - \mu_k^*) - \frac{I_1(\kappa_k^*)}{I_0(\kappa_k^*)} \right) = 0$$

We finally get:

$$\sum_{n=1}^{N} \eta_{nk} \ \cos(x_n - \mu_k^*) = S_k \frac{I_1(\kappa_k^*)}{I_0(\kappa_k^*)} \quad \square$$

Unfortunately we cannot derive a formula for κ_k^*, but it is possible to estimate it by numerically solving the previous equation for κ_k^*.

3.4 2-components Von Mises mixture

Using the previous formulas, we have implemented the expectation-maximization algorithm for a mixture of Von Mises.

Below are the results obtained fitting a 2 and 3 components von Mises mixture to our angular data.

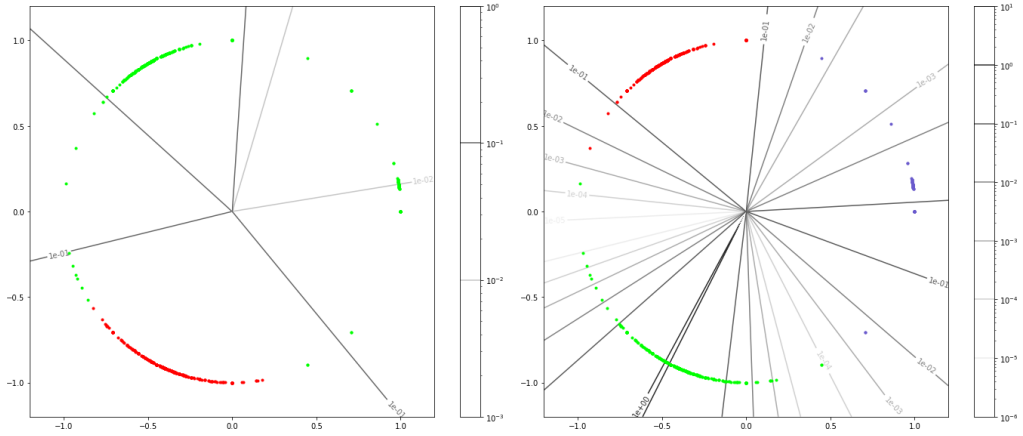


Figure 11: Angular data on the unit circle and fit Von-Mises mixture (left: 2 components, right: 3 components)

As expected, the contour levels of the Von Mises PDF are radial. The estimated parameters of the mixtures are as follows.

2-component parameters:

$$\boldsymbol{\kappa} = (21.95, 2.33)$$
$$\boldsymbol{\mu} = (-1.95, 1.90)$$
$$\boldsymbol{\pi} = (0.47, 0.53)$$

3-component parameters:

$$\boldsymbol{\kappa} = (34.49, 17.60, 17.89)$$
$$\boldsymbol{\mu} = (2.05, 0.14, -1.96)$$
$$\boldsymbol{\pi} = (0.42, 0.09, 0.48)$$

In the figure 18 of the appendix, we also give the histogram of the angular data fit using the estimated 3-component mixture with the parameters above.

4 Consistent estimators

4.1 Definition, state-of-the art and zoom on a specific approach

4.1.1 Preliminaries

Mixture models have seen numerous applications, especially specific and efficient ones such as mixtures of Poisson, binomials or Gaussians. There exists a huge body of literature dedicated to computational issues and theoretical aspects of mixture models when the number of components is known. However, we can face problems when the number of components is unknown in nearly any application, thus making the estimation of the number of components in a mixture model (or mixture complexity) an area of intense research effort.

We will first briefly discuss the two ways to apprehend this problem (Frequentist, Bayesian) before diving into a specific method which produces a consistent estimator for Gaussian mixture models. We say that an estimator $\hat{\beta}_n$ of a quantity of interest β is consistent if $\lim_{n\to\infty} \hat{\beta}_n = \beta$, i.e. it is guaranteed to converge to the true value. We now formalize this definition for consistent estimators of the number of components in a mixture model.

Definition 4.1. Let $(X_i)_{1\leq i\leq n}$ be a set of i.i.d. random variables which form a β-components mixture model. We say that $\hat{\beta}_n$ is a consistent estimator for β if and only if:

$$\forall \epsilon > 0, \lim_{n\to\infty} \mathbb{P}(|\hat{\beta}_n - \beta| \geq \epsilon) = 0.$$

This is the frequentist approach to parameter estimation. However, the number of components being unknown, it seems more adequate to consider the number of components as an additional unknown parameter. This makes the Bayesian approach attractive as it will treat the number of components in our mixture model as an additional unknown parameter that is to be estimated simultaneously with the others [1].

We now define posterior consistency as in [2]. Let us consider Θ a parameter space and denote m one of its elements, Π a prior distribution, i.e. a probability measure on Θ. $\{f_\theta : \theta \in \Theta\}$ then denotes a family of density functions parameterized by Θ. We note P_θ the probability distribution generated by f_θ. For any measurable subset A of Θ, the posterior distribution $\Pi(A|X_1, ..., X_n)$ is given by:

$$\Pi(A|X_1, ..., X_n) = \frac{J_A(X_1, ..., X_n)}{J_\Theta(X_1, ..., X_n)}$$

$$\text{where } J_\mathcal{X}(X_1, ..., X_n) = \int_\mathcal{X} \frac{f_\theta^{(n)}}{f_m^{(n)}}(X_1, ..., X_n)\Pi(d\theta)$$

Finally, let P_m^∞ stand for the joint distribution $\{X_i\}_{i=1}^\infty$ where m is the unknown number of components we are trying to estimate.

Definition 4.2. With Π a prior and $\Pi(\cdot|X_1, ..., X_n)$ a sequence of posterior distributions, we say our sequence of posteriors is a consistent estimator at m if:

$$\forall U \; neighborhood \; of \; m, P_m^\infty(\{\Pi(U|X_1, ..., X_n)\}) \to 1 \; almost \; surely.$$

4.1.2 State of the art

In the following, we will discuss both frequentist and bayesian approaches to estimate the number of components. In the frequentist framework, a natural approach to estimate the number of components of a mixture model would be based on a likelihood ratio test. Unfortunately, it appears [1] that the classic setting does not ensure enough conditions for the maximum likelihood estimates to hold. To enforce regularity conditions in this setting, many modifications of the likelihood ratio test were proposed. For example, [3] ensures a consistent estimator with penalized maximum likelihoods when the number of components is bounded and with various strong assumptions on the mixtures. More recently, [4] uses weaker conditions to ensure a similar consistent estimator. Recent work [5] proposes a bootstrap approach to this problem which gives satisfying results on Poisson mixtures. Overall, statistical tests are rarely used in our case as they are often numerically inefficient. Hence, it is often preferred to optimize penalized log-likelihood or resort to Bayesian methods.

From a Bayesian perspective, a commonly used approach consists in using information criteria to consistently estimate the number of components in a mixture model. This notion is based on penalizing the log-likelihood of a mixture model with m components. The BIC (Bayesian Information Criterion) [6] can be consistent in specific conditions but not in our setting, where it faces the same issues as the likelihood ratio test. However, the BIC has been shown to be consistent under various assumptions such as having bounded probability density functions for each component of the mixture [7]. Still, it suffers from not being asymptotically optimal (it often overestimates the number of components).

Information criterion that are computed using MCMC (Markov Chain Monte Carlo) methods such as the DIC (Deviance Information Criterion) [8] gained popularity recently because of their cost-efficiency but they face similar issues. Using marginal likelihood (for which the BIC is initially an approximation) approximated by simulation-based methods is also a promising method to consistently estimate the number of components in a mixture model, see [9] for example. Finally, a recent concept, sparse finite mixtures, introduced in [10], allows to solve the consistent estimator problem for mixture models. The basic idea is to first deliberately use too many components then apply a MCMC sampling procedure to estimate the unknown number of non-empty data clusters which can coincide under proper assumptions with the desired consistent estimator.

4.1.3 A sparse finite mixtures based consistent estimator

In the previous section, we evoked consistent estimators based on sparse finite mixtures and the general idea behind it, as introduced in [10]. We will now describe more thoroughly their approach as it provides a novel and numerically efficient way to compute a consistent estimator of the number of components in a Bayesian model-based clustering framework, which we consider to be more attractive because we have seen that as it treats the number of components as an unknown parameter, it is more adapted to truly estimate it.

In the context of model-based clustering, we assume the data to be generated from a finite set of mixture components, each corresponding to a specific cluster. Specifically, an observation y_i is drawn from :

$$f(y_i|\theta_1, ..., \theta_K, \eta) = \sum_{k=1}^{K} \eta_k f_k(y_i|\theta_k) \tag{2}$$

where η corresponds to the components weights and each $f_k(y_i|\theta_k)$ is assumed to belong to a known distribution family (Poisson or Gaussian for example). Here, we will consider all of them to be multivariate Gaussian densities of parameters $\theta_k = (\mu_k, \Sigma_k)$ where μ_k is the k-component mean, Σ_k its covariance matrix. K corresponds to the deliberately overestimated number of components and K_{true} is the one we aim to estimate.

Moreover, we assume a prior on the mixture weights: $\eta \sim Dir(e_0, ..., e_0)$ a symmetric Dirichlet prior. It has been shown in other references such as [11] that the hyperparameter e_0 is one of the most influential in the setting where we overestimate the number of components and that it has to be chosen very small. Then, a Gamma hyperprior $\mathcal{G}(a, a \times K)$ is assumed on e_0, following recommendations from [12]. The variance of e_0, $(aK^2)^{-1}$, is therefore very sensitive to the parameter a and has to be chosen big enough in order not to allow large values of e_0 that are not desired as explained in [11].

After a MCMC sampling (which goal is to empty superfluous components), we derive the posterior distribution $\Pi(K_0 = h, y)$, with $h \in [\![1, K]\!]$. Let us denote K_0 the number of non-empty components in the MCMC output, $K_0^{(m)}$ its value at iteration m and $N_k^{(m)}$ the number of observations allocated to component k at iteration m, linked with the following equation:

$$K_0^{(m)} = K - \sum_{k=1}^{K} \mathbb{1}_{N_k^{(m)}=0}.$$

The posterior distribution is then estimated by the corresponding relative frequency. Then, using the posterior mode estimator $\tilde{K_0}$, we can estimate the number of clusters K_0, which happens to be the most frequent number of clusters visited during MCMC sampling.

4.2 Evaluation of the consistency of an estimator of the number of components

4.2.1 Description

In order to check the consistency of an estimator, we need mixture models for which we know the true number of components and an estimator to evaluate it. To verify convergence of the estimator, we need to simulate mixture models with increasing sample size. A consistent estimator will output closer values when applied to bigger increasing sample size, as definition 4.1 states. Moreover, to check the robustness of the estimator, we can make the mixture's parametric distributions vary as well as the true number of components and see how it is handled by the estimator.

In our implementation, we will evaluate the consistency of the BIC based estimator. The BIC criterion is used for model selection: among a finite set of models, this criterion chooses the model of smallest BIC. Below is a definition of the Bayesian Information Criterion, which approximates the marginal likelihood of a model independently to the prior.

Definition 4.3. Let us consider the likelihood function $L(\theta, K)$ of a mixture model with K components for n observations $\{y_i\}_{i=1}^n$:

$$L(\theta, K) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \eta_k f_k(y_i|\theta_k) \right], \tag{3}$$

Denoting $\hat{\theta}_K$ as the maximum likelihood estimator corresponding to $L(\theta, K)$, we define the BIC as:

$$\text{BIC}(K) = -2 \log L(\hat{\theta}_K, K) + \alpha \log(n), \tag{4}$$

where α is the number of free parameters in the mixture model. The number of components selected by the BIC criterion is therefore:

$$\hat{K} = \underset{K \in \Theta}{\arg\min}\ \text{BIC}(K) \tag{5}$$

Where Θ is the parameter space to explore. To check the consistency of our BIC estimator, we will first compute the BIC criterion for model of various number of components and select the one with the lowest $\hat{K}$. Doing this for samples of increasing size, a consistent estimator should approach the true number of components as n grows.

4.2.2 Experiments

To test our simple method, we simulate samples of increasing size from a 4-components GMM, and we estimate the optimal K using the previously described method for each sample.

In figure 12 below is the BIC-estimated value of K as a function of the size n of the dataset on which it is estimated:

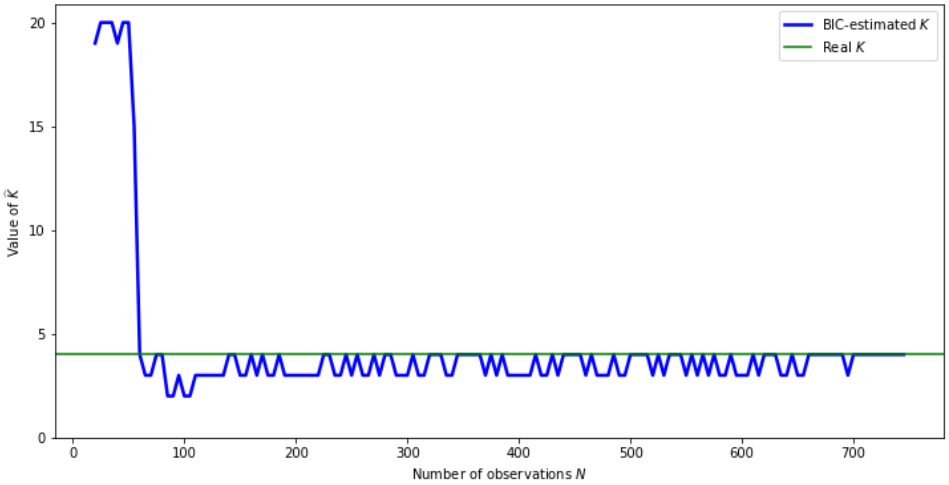


Figure 12: Estimated $\widehat{K}$ as a function of n

We can see that the BIC-estimated value of K seems to converge quickly to its true value (indicated in green). We also compare this criterion to the Akaike information criterion (AIC) in figure 19 of the annex: this comparison is partly unfair as the latter does not converge in probability to the true model parameters.

5 Implementation of Von Mises mixtures

5.1 Sampling and PDF function

Implementing the PDF of the Von Mises distribution function is straight forward, using its given formula.
To write a sampling function from any distribution $\mathcal{D}$, given that we have access to its cumulative distribution function $F(x)$, the following property can be used. For $X \sim D$, we have:

$$F(X) = U,\ U \sim \mathcal{U}_{[0,1]}$$

Using the quantile function of this distribution, we thus have by inversion:

$$F^{-1}(U) = X,\ X \sim \mathcal{D}$$

In our case, we do not have access to this quantile function, as the CDF of Von Mises does not have an analytic form: this method cannot be used directly.

Instead, we can make use of rejection sampling schemes. Here, we implement the algorithm proposed in [13] to sample from a Von Mises distribution using the ratio-of-uniforms [14] method.
In figure 13 below is a normalized histogram of 500 points sampled using our function, matched with the corresponding Von Mises PDF:

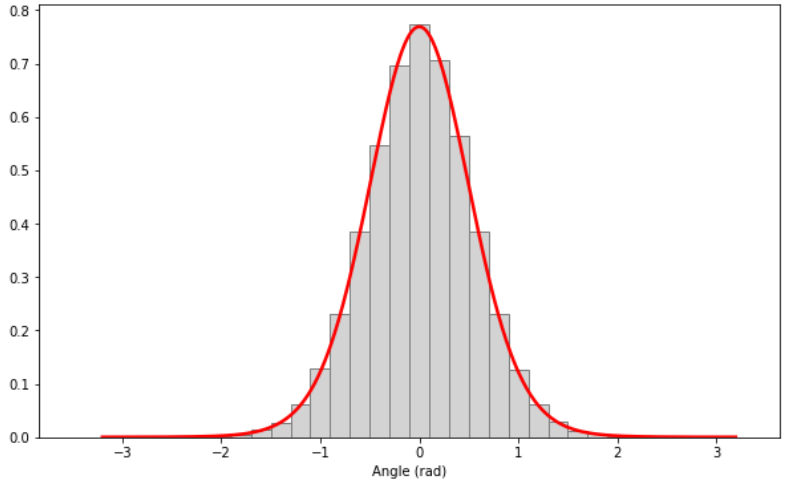


Figure 13: Sampled Von-Mises points and Von-Mises PDF

In order to assess the quality of our sampling method, and since Von-Mises is a continuous distribution, we can use the Kolmogorov-Smirnov test to compare the distribution of our generated samples with the Von Mises PDF.
With $F_n(x)$ the empirical distribution function obtained from generated samples, and $F(x|\mu, \kappa)$ the CDF of the Von-Mises distribution, this statistic is defined as:

$$D_n = \sup_x |F_n(x) - F(x|\mu, \kappa)|$$

From this statistic, the null hypothesis of two identical distributions is rejected at level α if $\sqrt{n}D_n > K_\alpha$, where K_α is a quantile of the Kolmogorov distribution.

In the table below are the p-values obtained for various κ parameters, with $n = 10^4$ samples for each test:

κ values	0.5	10	50	100
p-value	0.69	0.25	0.88	0.90

Our p-values are high enough in all of the tests to not reject the null hypothesis (identical distributions), which confirms that our samples follow the desired distribution.

To sample from a mixture of Von Mises distributions, we simply follow the method previously described in section 1.2 to sample from a mixture.

5.2 3-components mixture simulation

We simulate a 3-components mixture with the following arbitrary parameters:

$$\begin{aligned}\boldsymbol{\kappa} &= (15, 10, 60)\\ \boldsymbol{\mu} &= (-\frac{2}{3}\pi, 0, \frac{2}{3}\pi)\\ \boldsymbol{\pi} &= (0.3, 0.25, 0.45)\end{aligned}$$

We obtain the following histogram, in figure 14 below, alongside the mixture PDF in red.

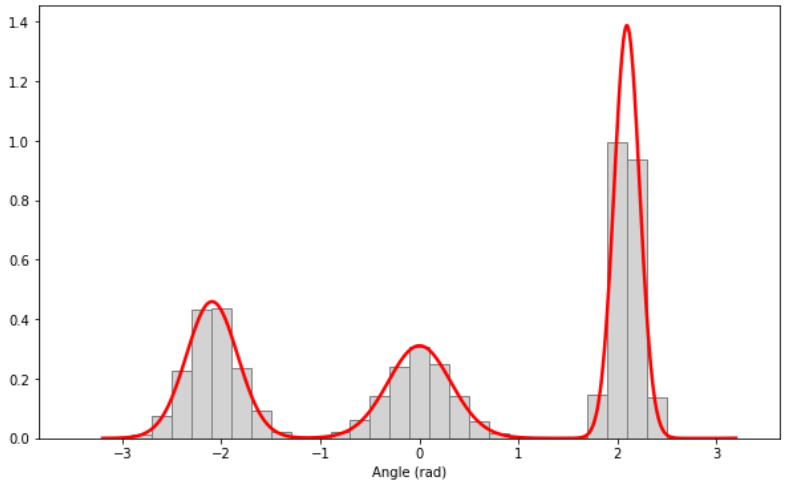


Figure 14: Sampled mixture of Von-Mises points and mixture PDF

Converting angles to points on the unit circle, we get the fit represented in figure 15 of the mixture model, with the contour levels of the mixture of Von Mises probability density function.

From the generated data, we can now try to reestimate mixture parameters using our previous implementation of expectation maximization for Von Mises mixtures.

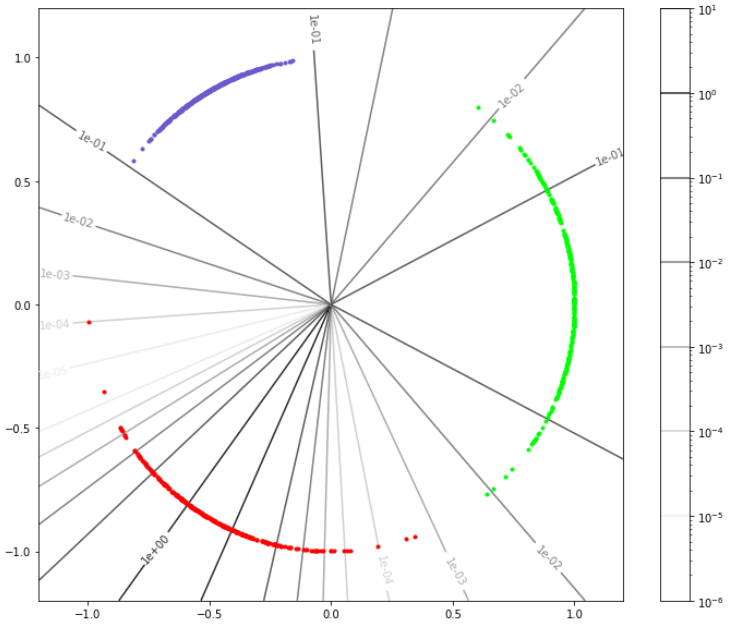


Figure 15: Sampled points from mixture of Von-Mises and mixture PDF contour levels

5.3 Parameters estimation from simulated data

Below in figure 16 is a histogram plot of our sampled data, alongside the PDF of the original distribution (green) and our EM-estimated distribution (red): the two distributions seem strikingly close.

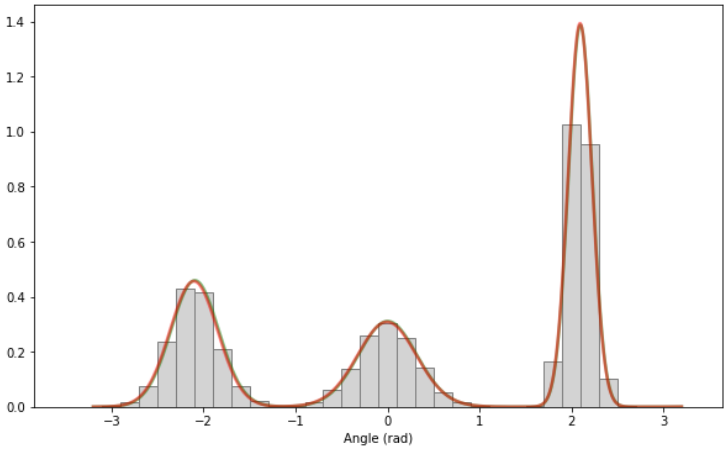


Figure 16: Histogram of sampled points, real PDF (green) and estimated PDF (red)

In order to evaluate the quality of the estimation, we simulate 10^3 points from the previous mixture and run the EM algorithm. Results are given in the following table.

	real	**estimated (rounded)**	**average error (%)**
$\boldsymbol{\pi}$	$(0.3, 0.25, 0.45)$	$(0.29, 0.25, 0.45)$	0.32
$\boldsymbol{\kappa}$	$(15, 10, 60)$	$(14.97, 9.82, 60.13)$	0.71
$\boldsymbol{\mu}$	$(-\frac{2}{3}\pi, 0, \frac{2}{3}\pi)$	$(-2.10, 0, 2.09)$	0.24

This numerical assessment confirms the visual observation that the estimated parameters are very close to the real values.

Additionally, in the figure 20 of the appendix, we also compare the simulated 3-components mixture with the estimated parameters on a bivariate plot. We can see again that the estimation is very accurate, as we recover almost exactly the same clustering and contour levels.

6 Appendix

6.1 Additional figures

Adding an additional component to the Gaussian Mixture Model of section 2.2 leads to the following mixture PDF and predicted labels:

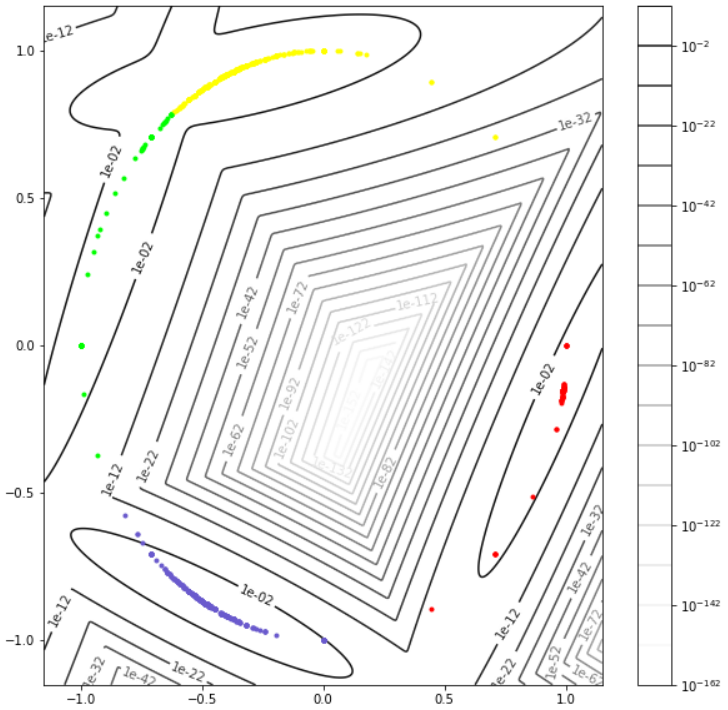


Figure 17: Predictions and PDF contour levels for a GMM with $K = 4$

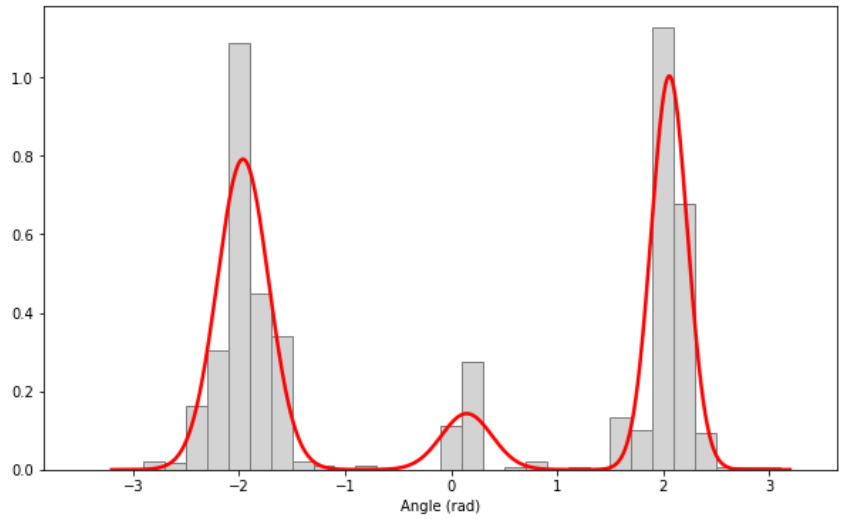


Figure 18: Histogram of the angular data fit by 3-component Von Mises mixture

The Akaike Information Criterion (AIC) is defined as follows:

$$\mathrm{AIC}(K) = -2 \log L(\hat{\theta}_K, K) + 2\alpha$$

Compared to BIC, instead of regularizing with a function of the number of observations, we only take into account the number of degrees of freedom in our mixture.

Below is a comparison of the two criterions in the estimation of the number of components of a Gaussian Mixture Model, for increasing sample sizes:

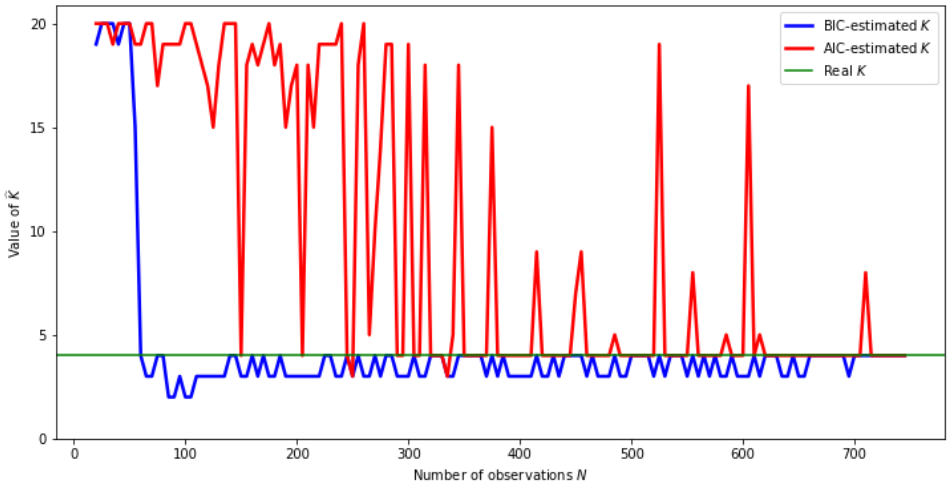


Figure 19: BIC and AIC estimations of the number of components of a GMM

Comparing simulated data with the clustering resulting from a mixture with parameters from the EM algorithm leads to the following figure:

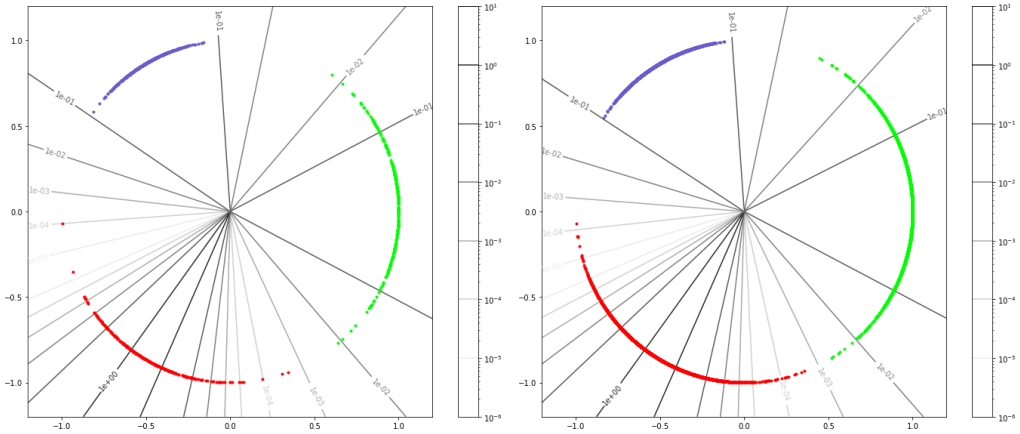


Figure 20: Simulated (left) and reconstructed (right) Von Mises clustering

References

[1] Gilles Celeux, Sylvia Fruewirth-Schnatter, and Christian P. Robert. Model selection for mixture models - perspectives and strategies, 2018.

[2] Taeryon Choi and R. V. Ramamoorthi. *Remarks on consistency of posterior distributions*, volume Volume 3 of *Collections*, pages 170–186. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2008. doi: 10.1214/074921708000000138. URL https://doi.org/10.1214/074921708000000138.

[3] Elisabeth Gassiat. Likelihood ratio inequalities with applications to various mixtures. *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, 38(6): 897 – 906, 2002. ISSN 0246-0203. doi: https://doi.org/10.1016/S0246-0203(02)01125-1. URL http://www.sciencedirect.com/science/article/pii/S0246020302011251.

[4] E. Gassiat and R. van Handel. Consistent order estimation and minimal penalties. *IEEE Transactions on Information Theory*, 59(2):1115–1128, Feb 2013. ISSN 1557-9654. doi: 10.1109/tit.2012.2221122. URL http://dx.doi.org/10.1109/TIT.2012.2221122.

[5] Peter Schlattmann. On bootstrapping the number of components in finite mixtures of poisson distributions. *Statistics and Computing*, 15(3):179–188, July 2005. ISSN 0960-3174. doi: 10.1007/s11222-005-1307-8. URL https://doi.org/10.1007/s11222-005-1307-8.

[6] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[7] Christine Keribin. Consistent estimation of the order of mixture models. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 49–66, 2000.

[8] CP Robert. Discussion on" bayesian measures of model complexity and fit"(by spiegelhalter, dj et al.). *Journal of the Royal Statistical Society, Ser. B*, 64:621–622, 2002.

[9] Nicolas Chopin and Christian P Robert. Properties of nested sampling. *Biometrika*, 97(3):741–755, 2010.

[10] Gertraud Malsiner-Walli, Sylvia Frühwirth-Schnatter, and Bettina Grün. Model-based clustering based on sparse finite gaussian mixtures. *Statistics and computing*, 26(1-2):303–324, 2016.

[11] Judith Rousseau and Kerrie Mengersen. Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):689–710, 2011.

[12] Hemant Ishwaran, Lancelot F James, and Jiayang Sun. Bayesian model selection in finite mixtures by marginal density decompositions. *Journal of the American Statistical Association*, 96(456):1316–1332, 2001.

[13] Lucio Barabesi. *Generating von Mises variates by the ratio-of-uniforms method*, pages 417–426. Statistica Applicata, 1995. URL http://sa-ijas.stat.unipd.it/sites/sa-ijas.stat.unipd.it/files/417-426.pdf.

[14] Luca Martino, David Luengo, and Joaquín Míguez. *Ratio of Uniforms*, pages 159–196. Springer International Publishing, Cham, 2018. ISBN 978-3-319-72634-2. doi: 10.1007/978-3-319-72634-2_5. URL https://doi.org/10.1007/978-3-319-72634-2_5.