Fundamentals of Probabilistic Data Mining Lab 1

Zeinab Abdallah, Lucas Batier, Fanny Lehmann, Predrag Pilipovic

November 27, 2019

Note. Code for this Lab is available in the notebook there:

`https://colab.research.google.com/drive/197YWO_C5k86bvoaiCJHxd4rp12fTdE6B`

1 Mixture Models

The Unistroke alphabet, closely related to Graffiti[1], is an essentially single-stroke shorthand handwriting recognition system used in PDAs. The data set is composed of 50×6 time-trajectories representing the drawing of letters A, E, H, L, O and Q in a plane.

Here you will focus on modelling letter A (actually drawn as a Λ). After some pre-processing, we obtain the data set "Amerge.txt" (you can find it in the zip file), which is composed of every stroke of every trial for the gestures associated with that letter (the temporal aspect of sequences and the separations between sequences were lost here).

1.1 Lab work

1.1.1 Preparatory work and modelling

Do this before the class. Questions about this part will be answered only at the beginning of the practical session.

1. **Prove the reestimation formula for Gaussian Mixture Model (GMM) (exercise 2 in the slides).**

   Let $(X_1, X_2, ..., X_n)$ be a discrete simple random sample from the mixture distribution. Let K be the number of mixture components, $Z = (Z_1, Z_2, ..., Z_n)$ latent variable representing the mixture component for X_i, such that $Z_i \in \{1, 2, ..., K\}$, for all $i = 1, 2, ..., n$. We call $P(X_i = x \mid Z_i = k)$ the mixture component and π_k the mixture proportion or the probability that X_i belogns to the k-th mixture component. Now, we have mass function of the mixture distribution given by

   $$P(X_i = x) = \sum_{k=1}^{K} \pi_k P(X_i = x \mid Z_i = k).$$

   Now, let $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n)$ be from multivariate Gaussian mixture model, i.e.

   $$\mathbf{X}_i \mid Z_i = k \sim \mathcal{N}(\mu_k, \Sigma_k).$$

   Then, the probability density function for $\mathbf{X}_i$ will be given by

   $$p_{\mathbf{X}_i}(\mathbf{x}) = \sum_{k=1}^{K} P(Z_i = k)\mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k).$$

[1] `http://en.wikipedia.org/wiki/Graffiti_(Palm_OS)`

At this point we do not know parameters $\theta = (\mu_1, ..., \mu_K, \Sigma_1, ..., \Sigma_K, \pi_1, ..., \pi_K)$. We will estimate them by EM algorithm. Firstly, let us can compute $P(Z_i = k \mid \mathbf{X}_i, \theta^{(t)})$, where $\theta^{(t)}$ is estimation of parameters θ at time t. So we have

$$\begin{aligned} P(Z_i = k \mid \mathbf{X}_i = \mathbf{x}_i, \theta^{(t)}) &= \frac{p_{\mathbf{X}_i|Z_i=k}(\mathbf{x}_i) \cdot P(Z_i = k)}{p_{\mathbf{X}_i}(\mathbf{x}_i)} \\ &= \frac{\pi_k \cdot \mathcal{N}\left(\mathbf{x}_i; \mu_k^{(t)}, \Sigma_k^{(t)}\right)}{\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}\left(\mathbf{x}_i; \mu_k^{(t)}, \Sigma_k^{(t)}\right)} = \gamma_{Z_i}^{(t)}(k). \end{aligned}$$

So, we defined $\gamma_{Z_i}^{(t)}(k)$ in previous line. In order to perform EM algorithm we need to calculate $\log p(\mathbf{x}, Z \mid \theta)$. We know that

$$p(\mathbf{x}, Z \mid \theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\pi_k \cdot \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k)\right)^{I(Z_i=k)},$$

so we have

$$\log p(\mathbf{x}, Z \mid \theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} I(Z_i = k)(\log \pi_k + \log \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k)).$$

Now we can compute $Q\left(\theta, \theta^{(t)}\right)$ which we will maximize in order to have update rule from the EM algorithm.

$$\begin{aligned} E_{Z|\mathbf{X}=\mathbf{x},\theta^{(t)}}[\log p(\mathbf{x}, Z \mid \theta)] &= E_{Z|\mathbf{X}=\mathbf{x},\theta^{(t)}} \left[\sum_{i=1}^{n} \sum_{k=1}^{K} I(Z_i = k)(\log \pi_k + \log \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k))\right] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} E_{Z|\mathbf{X}_i=\mathbf{x}_i,\theta^{(t)}}[I(Z_i = k)](\log \pi_k + \log \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k)) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} P(Z_i = k \mid \mathbf{X}_i = \mathbf{x}_i, \theta^{(t)})(\log \pi_k + \log \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k)) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{Z_i}^{(t)}(k)(\log \pi_k + \log \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k)). \end{aligned}$$

Having in mind that density function of multivariate normal distribution is

$$\mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_k)},$$

where p is the dimension of $\mathbf{x}_i$. Now we have $Q\left(\theta, \theta^{(t)}\right)$, it is given by

$$Q\left(\theta, \theta^{(t)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{Z_i}^{(t)}(k) \left(\log \pi_k - \frac{p}{2} \log 2\pi - \frac{1}{2} \log |\Sigma_k| - \frac{1}{2}(\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_k)\right).$$

Let us, for the convenience, introduce a new variable

$$n_k^{(t)} = \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k).$$

Note here that we have a constraint $\sum\limits_{k=1}^{K} \pi_k = 1$, so we need Lagrange multiplier in order to maximize $Q\left(\theta, \theta^{(t)}\right)$. We need to maximize the Lagrangian function

$$\mathcal{L}\left(\theta, \theta^{(t)}, \lambda\right) = Q\left(\theta, \theta^{(t)}\right) + \lambda\left(1 - \sum_{i=1}^{n} \pi_k\right).$$

A first partial derivative of $\mathcal{L}$ in terms of π_k is

$$\frac{\partial \mathcal{L}}{\pi_k} = \sum_{i=1}^{n} \frac{\gamma_{Z_i}^{(t)}(k)}{\pi_k} - \lambda = 0,$$

which leads us to the update rule $\pi_k^{(t+1)} = \dfrac{n_k^{(t)}}{\lambda}$. To calculate λ, we use

$$1 = \sum_{k=1}^{K} \pi_k^{(t+1)} = \sum_{k=1}^{K} \frac{n_k^{(t)}}{\lambda},$$

which means that

$$\begin{aligned}\lambda &= \sum_{k=1}^{K} n_k^{(t)} = \sum_{k=1}^{K}\sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k) \\ &= \sum_{i=1}^{n} \underbrace{\sum_{k=1}^{K} P(Z_i = k \mid \mathbf{X}_i = \mathbf{x}_i, \theta^{(t)})}_{1} = n.\end{aligned}$$

This means we have $\pi_k^{(t+1)} = \dfrac{n_k^{(t)}}{n}$. In order to find update rules for other parameters we need to solve next system

$$\begin{cases} \nabla_{\mu_k}\mathcal{L} = \sum\limits_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\Sigma_k^{-1}(\mathbf{x}_i - \mu_k) = 0 \\ \nabla_{\Sigma_k}\mathcal{L} = \sum\limits_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\left(-\dfrac{1}{2}\Sigma_k^{-1} + \dfrac{1}{2}\Sigma_k^{-2}(\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T\right) = 0. \end{cases}$$

From the first equation we have

$$\Sigma_k^{-1}\sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\mathbf{x}_i = \Sigma_k^{-1}\mu_k \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k),$$

which leads us to

$$\mu_k^{(t+1)} = \frac{1}{n_k^{(t)}} \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\mathbf{x}_i.$$

Putting the previous solution into the second equation we have

$$\Sigma_k^{-1} \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k) = \Sigma_k^{-2} \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)(\mathbf{x}_i - \mu_k^{(t+1)})(\mathbf{x}_i - \mu_k^{(t+1)})^T,$$

which means that

$$\Sigma_k^{(t+1)} = \frac{1}{n_k^{(t)}} \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k) \left(\mathbf{x}_i - \mu_k^{(t+1)}\right) \left(\mathbf{x}_i - \mu_k^{(t+1)}\right)^T.$$

2. **Simulate a sample of size 500 of the following bivariate GMM:**

$$0.3 \cdot \mathcal{N}(\mu_1, \Sigma_1) + 0.7 \cdot \mathcal{N}(\mu_2, \Sigma_2)$$

with

$$\mu_1 = \begin{pmatrix} -3 \\ 0 \end{pmatrix}, \mu_2 = \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \text{ and } \Sigma_1 = \begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 5 & 2 \\ 2 & 2 \end{pmatrix}.$$

Hint: `numpy.random.multivariate_normal`.

Plot the synthetic data set and check if it corresponds to figures in the slides of the class (Page 6).

In order to simulate data, first of all, we choose classes for each point randomly, which means: generating the data for Z from binomial distribution $\mathcal{B}(n, p)$, where p is probability $P(Z = 1)$. Now we have the classes, so for all the class we generated the numbers from the appropriate Gaussian distribution.

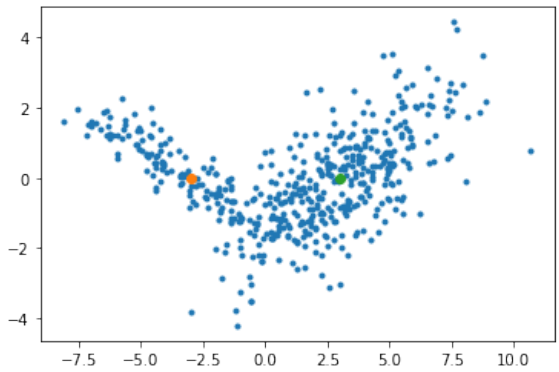


Figure 1: GMM

3. **Download (from `chamilo`), load and plot the Unistroke data set (letter A) and provide the figure.**

We downloaded, loaded and plot the Unistroke data set of letter A (Amerge.txt) rotated by $\frac{\pi}{2}$ and provide it in Figure 2.

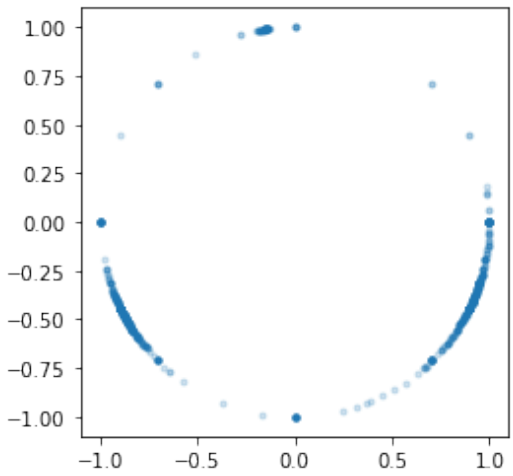


Figure 2: Unistroke data set for letter A

4. **Do you think a 2-components GMM could be appropriate for letter A? Why?**

   We used a transparency coefficient to show the density of points. We can see that there is 3, and not 2, groups which can be modeled by a GMM. This seems logical because we need 3 points to draw the letter A like this: Λ.

1.1.2 Data analysis: Gaussian model

1. **Estimate a bivariate GMM on the letter A data set and provide the estimated parameters.**

   For the parameters we got next parameters

$$\mu_1 = \begin{pmatrix} 0.76 \\ -0.20 \end{pmatrix}, \mu_2 = \begin{pmatrix} -0.84 \\ -0.48 \end{pmatrix}, \text{ and } \Sigma_1 = \begin{pmatrix} 0.14 & -0.14 \\ -0.14 & 0.23 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 0.033 & -0.027 \\ -0.026 & 0.033 \end{pmatrix}.$$

   and weights

$$\pi = (0.47, 0.53).$$

   We notice that both weights are similar. This is what we expected as the two "feet" of the letter Λ contains approximately the same number of points.

2. **Label the data using the estimated model and show the pdf of the estimated GMM. (Provide one figure with the data labeled in color overlapping on the contours of the log(pdf), please add inline labels for the contours).**

   Hint: `mixture.GaussianMixture.predict`, `numpy.meshgrid`.

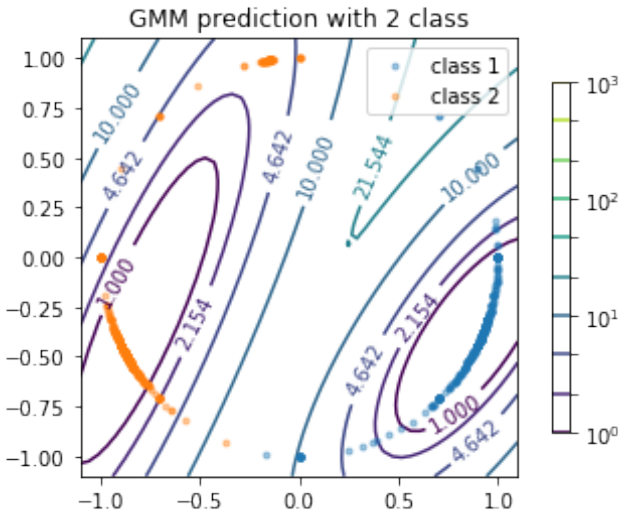


Figure 3: Label data for the bivariate GMM

3. **To validate the assumption of bivariate Gaussian mixture:**

   (a) **Plot each marginal histogram (in x and y) and add the estimated mixture of univariate Gaussian pdfs to the figure.**

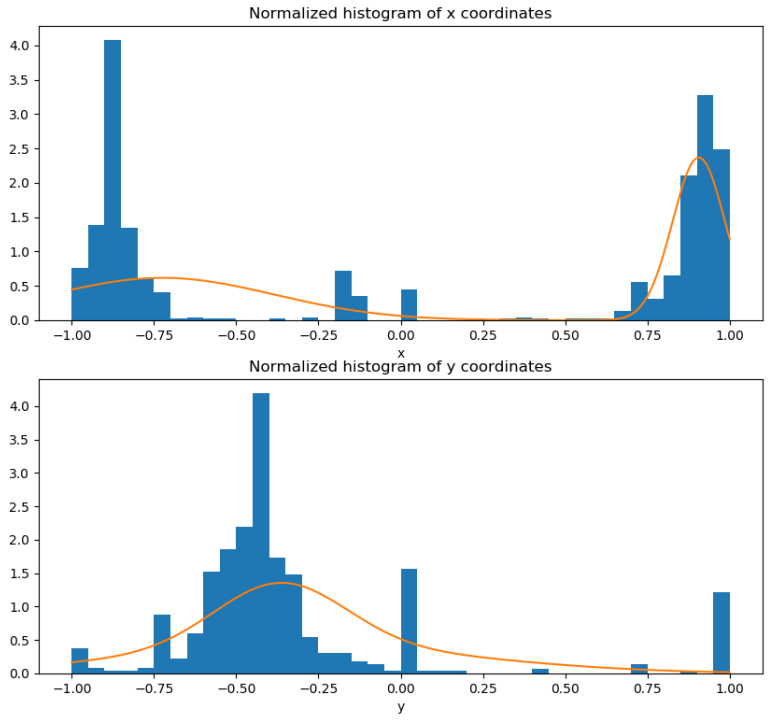


Figure 4: Histogram and density of each coordinate from a bivariate GMM

We clearly see 3 groups for the x coordinate and then cannot be caught by a 2 component GMM. That is why the lowest mean is higher than the mean of the lowest x histogram. Moreover, both "feet" of the letter Λ have almost the same y coordinate, leading to a single group on the y histogram.

(b) **For each marginal, provide separate histograms of each cluster and add the estimated univariate Gaussian pdf to the figure.**

Hint: `scipy.stats.norm`.

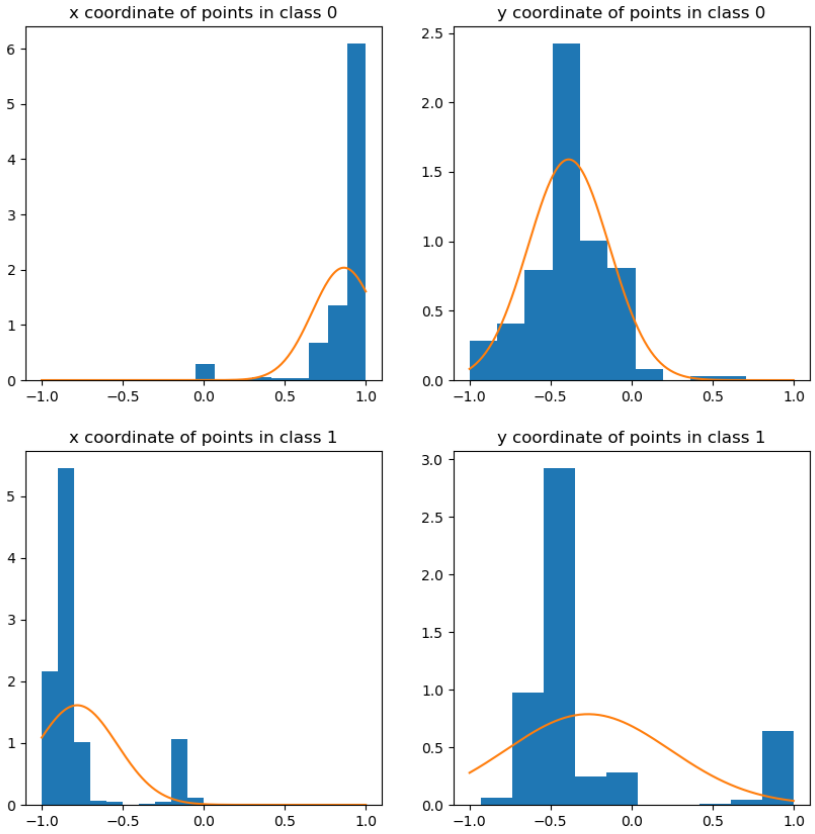


Figure 5: Histogram of each coordinate by cluster

We notice that histograms are close from the density functions for points in class 0. However, as the class 1 contains point at the top of the letter Λ, the density function is influenced by the outlier component.

4. **Comment the results of questions 3 (a) and (b). What to think about the bivariate Gaussian mixture assumption? Why?**

From the beginning we provided explanations why is bivariate GMM bad model for the letter A. The problem is we actually need 3 clusters and not to in order to model three points of letter A. On the contour plot in question 2 (Figure 3) we can see an influence

from the top group on the left group, so the ellipse does not follow the pattern of data. We can compare this plot with plot when we have 3 clusters (Figure 6).

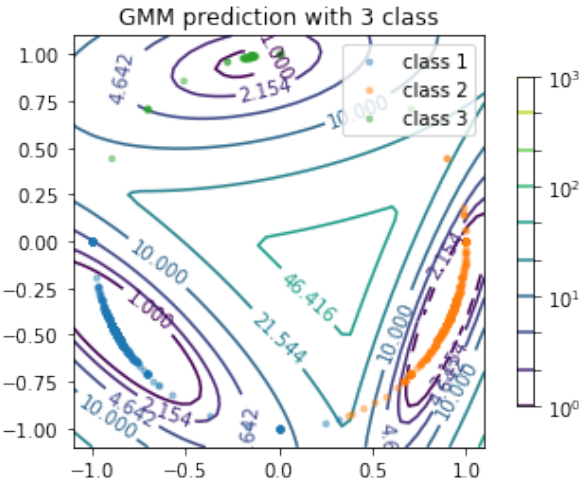
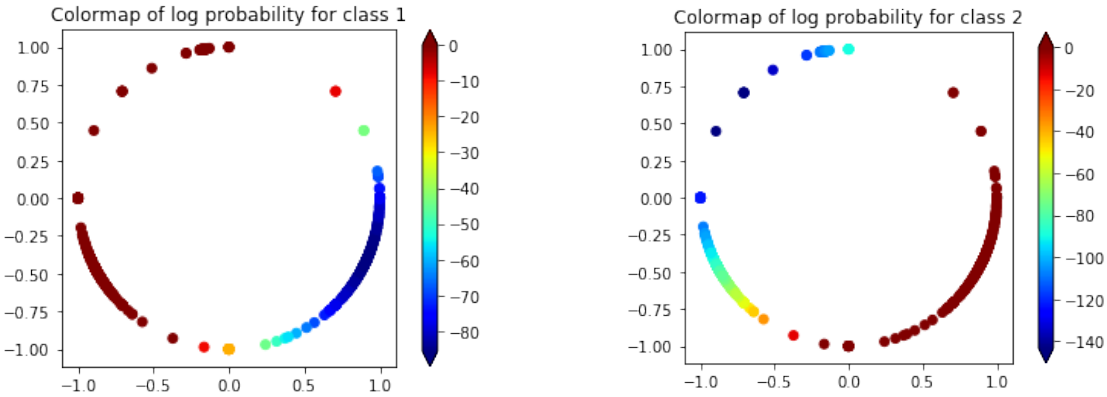


Figure 6: Label data for the trivariate GMM

Now, we can see that contours explain data much better than with bivariate model. So, our conclusion is that we need 3 but not 2 clusters to model letter A.

5. **Plot each data point x_i with some colourmap corresponding to $P(Z_i = 1 \mid X_i)$ (you may plot $\log P(Z_i = 1 \mid X_i)$ instead). How to interpret that plot?**



(a) $\log P(Z_i = 1 \mid X_i)$ (b) $\log P(Z_i = 0 \mid X_i)$

Figure 7: Data with colourmap corresponding to $\log P(Z_i \mid X_i)$

Those plots show the probability of each points to belong to the first or the second class by using a colormap. We used logarithm of the probability to smooth the color gradient. We can notice that unlike k-means we do not have hard boundaries on classes, instead for every point we have a probability.

1.2 Mandatory additional questions

The aim of this part is to compare mixture of von Mises distributions with Gaussian mixtures.

1. **Transform the Unistroke data to angular data. Plot the histogram of angles and comment.**

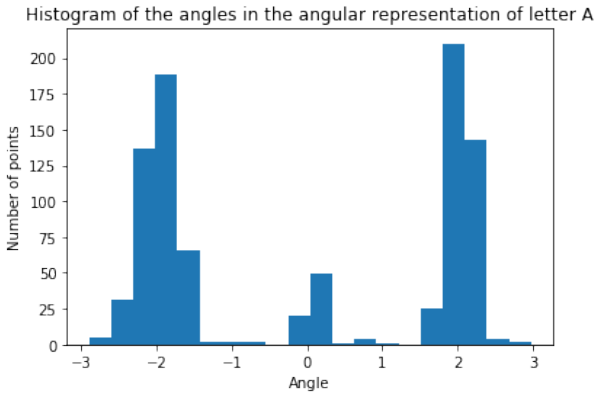


Figure 8: Angles histogram

We observe from figure 8 that only one dimension (the angle) is needed to catch the three groups we aim at reconstructing.

2. **Define von Mises and mixtures of von Mises distributions.**

Von Mises distribution is a continuous probability distribution on the circle (over 2π) it could be seen at the circular analogue of the normal distribution. The von Mises probability density function (**vM**) for a given angle θ is given by:

$$\mathbf{vM}(\theta \mid \mu, \kappa) = \frac{e^{\kappa \cos(\theta - \mu)}}{2\pi I_0(\kappa)}, \quad \mu \in \mathbb{R}, \kappa > 0,$$

where $I_0(\kappa)$ is the modified Bessel function of order zero, i.e.

$$I_0(x) = \frac{1}{\pi} \int_0^{\pi} e^{x \cos\theta} d\theta.$$

As far as the parameters concerned, $1/\kappa$ is analogous to the σ^2 and μ corresponds to the mean like in the Gaussian distribution.

3. **A priori, would a mixture of von Mises distributions be more or less adequate than Gaussian mixtures on the real data set of part 1.1? Why?**

Von Mises distributions will be more adequate than Gaussian mixtures because von Mises works on angular data and we can easily see on angles histogram (Figure 8) that we catch all the key information in one dimension instead of the Unistroke data where we need two dimensions.

4. **Provide equations for the E-step and M-step of the EM algorithm for mixtures of von Mises distributions. Justify these results with formal computations.**

   First part of this computation will be the same as for the Gaussian Mixture Model. Let $\Phi = (\Phi_1, \Phi_2, ..., \Phi_n)$ be simple random sample from the von Mises Mixture distribution, i.e.

   $$\Phi_i \mid Z_i = k \sim \mathbf{vM}(\mu_k, \kappa_k).$$

   Then, the probability density function for φ_i will be given by

   $$p_{\Phi_i}(\varphi) = \sum_{k=1}^{K} P(Z_i = k)\mathbf{vM}(\varphi; \mu_k, \kappa_k) = \sum_{k=1}^{K} \pi_k \cdot \mathbf{vM}(\varphi; \mu_k, \kappa_k).$$

   At this point we do not know parameters $\theta = (\mu_1, ..., \mu_K, \kappa_1, ..., \kappa_K, \pi_1, ..., \pi_K)$. We will estimate them by EM algorithm. Firstly, let us compute $P(Z_i = k \mid \Phi_i, \theta^{(t)})$, where $\theta^{(t)}$ is the estimation of parameters θ at time t. So we have

   $$\begin{aligned} P(Z_i = k \mid \Phi_i = \varphi_i, \theta^{(t)}) &= \frac{p_{\Phi_i|Z_i=k}(\varphi_i) \cdot P(Z_i = k)}{p_{\Phi_i}(\varphi_i)} \\ &= \frac{\pi_k \cdot \mathbf{vM}\left(\varphi_i; \mu_k^{(t)}, \kappa_k^{(t)}\right)}{\sum_{k=1}^{K} \pi_k \cdot \mathbf{vM}\left(\varphi_i; \mu_k^{(t)}, \kappa_k^{(t)}\right)} = \gamma_{Z_i}^{(t)}(k). \end{aligned}$$

   So, we defined $\gamma_{Z_i}^{(t)}(k)$ in previous line. In order to perform EM algorithm we need to calculate $\log p(\varphi, Z \mid \theta)$. We know that

   $$p(\varphi, Z \mid \theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\pi_k \cdot \mathbf{vM}(\varphi_i; \mu_k, \kappa_k)\right)^{I(Z_i=k)},$$

   so we have

   $$\log p(\varphi, Z \mid \theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} I(Z_i = k)(\log \pi_k + \log \mathbf{vM}(\varphi_i; \mu_k, \kappa_k)).$$

   So, we have the same thing as for the GMM. From now one, we will skip calculation which are the same, and provide the result we have.

   (E) For the E step we need to determine $Q\left(\theta, \theta^{(t)}\right)$, and it will be

   $$\begin{aligned} Q\left(\theta, \theta^{(t)}\right) &= \mathbb{E}_{Z|\Phi=\varphi,\theta^{(t)}}[\log p(\varphi, Z \mid \theta)] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{Z_i}^{(t)}(k)(\log \pi_k + \log \mathbf{vM}(\varphi_i; \mu_k, \kappa_k)) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{Z_i}^{(t)}(k)(\log \pi_k + \kappa_k \cos(\varphi_i - \mu_k) - \log 2\pi - \log I_0(\kappa_k)). \end{aligned}$$

(M) For the M step we need to maximize function Q in terms of θ, to find an update rule. Like before, let us introduce a new variable

$$n_k^{(t)} = \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k).$$

To update π_k we do not need to know the distribution of Φ, so it will be the same as for the GMM, i.e.

$$\pi_k^{(t+1)} = \frac{n_k^{(t)}}{n}.$$

In order to find update rules for other parameters first we need to notice the next property of modified Bassel function of order zero

$$\frac{dI_0(x)}{dx} = I_1(x) = \frac{1}{\pi}\int_0^{\pi} \cos\theta e^{x\cos\theta} d\theta.$$

So, now we have a system

$$\begin{cases} \dfrac{\partial Q}{\partial \mu_k} = \sum\limits_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\kappa_k \sin(\varphi_i - \mu_k) = 0 \\ \dfrac{\partial Q}{\partial \kappa_k} = \sum\limits_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\left(\cos(\varphi_i - \mu_k) - \dfrac{I_1(\kappa_k)}{I_0(\kappa_k)}\right) = 0 \end{cases}$$

First equation we can divide by κ_k, because we presume $\kappa_k > 0$ for the **vM** distribution. After applying addition formula for sin, we have

$$\sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\left(\sin\varphi_i\cos\mu_k - \cos\varphi_i\sin\mu_k\right) = 0.$$

After dividing the last equation by $\cos\mu_k$ we have

$$\begin{aligned} &\sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\left(\sin\varphi_i - \cos\varphi_i\tan\mu_k\right) = 0 \\ \Leftrightarrow &\sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\sin\varphi_i - \tan\mu_k \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\cos\varphi_i = 0 \\ \Rightarrow &\tan\mu_k^{(t+1)} = \frac{\sum\limits_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\sin\varphi_i}{\sum\limits_{i=1}^{n} \gamma_{Z_i}^{(t)}(k)\cos\varphi_i}. \end{aligned}$$

Note here, that it would not be correct to find a general solution of the previous equation, because we need only one $\mu_k^{(t+1)}$. Reason for this is because our data must be in some interval of length 2π, and thus μ will be in the same interval. Also, we did not discuss the case when $\cos\mu_k = 0$. Then we will have

$$\mu_k = \frac{\pi}{2} + l\cdot\pi,\ l \in \mathbb{Z}.$$

Then, our first equation from the system would look like

$$\sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k) \cos \varphi_i = 0.$$

Now, if we put that in our update rule for μ_k we will have $\tan \mu_k^{(t+1)} \in \{-\infty, +\infty\}$, and that will exactly correspond to the μ_k. So, our rule for μ_k works in the general case. We need to find the rule for κ_k now. From the second equation in the system and the solution for $\mu_k^{(t+1)}$ we have

$$\frac{I_1\left(\kappa_k^{(t+1)}\right)}{I_0\left(\kappa_k^{(t+1)}\right)} = \frac{1}{n_k^{(t)}} \sum_{i=1}^{n} \gamma_{Z_i}^{(t)}(k) \cos\left(\varphi_i - \mu_k^{(t+1)}\right).$$

The last equation must be computed numerically in order to find the update rule for κ, because of the Bassel functions.

5. **Fit a 2-components mixture of von Mises distributions on the Unistroke data set of part 1.1. List the estimated parameters and color the data (in original form) by the estimated labels.**

   Hint: `You may find an existing python library for mixtures of von Mises.`

   The parameters we get are the following :

$$\mu = (2.08, -1.85) \quad \kappa = (102.67, 1.69) \quad \pi = (0.38, 0.62)$$

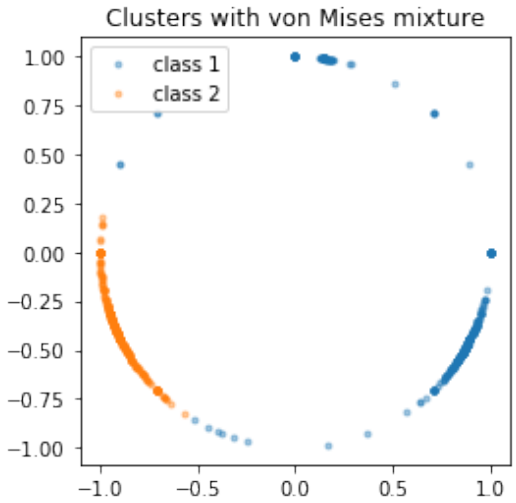


Figure 9: Label data with the 2 component Von Mises mixture

Again, we provide it for 3 class. In Figure 10 we can see that the green class is all over the circle which suggests that κ_{green} is really small (which corresponds to really big σ^2

in Gaussian case). Similarly, for the plot in Figure 9 we see that blue class is rather all over the circle, and corresponded κ is 1.69.

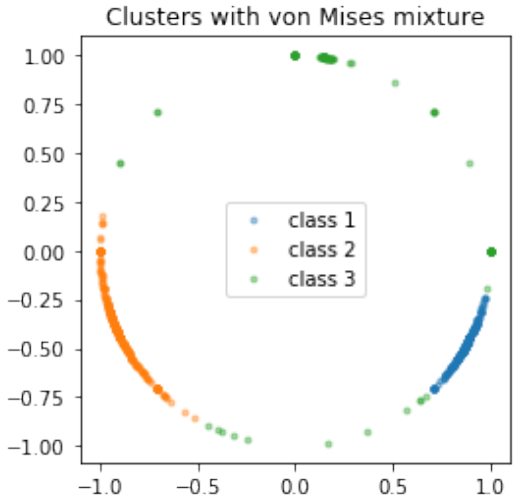


Figure 10: Label data with the 3 component Von Mises mixture

1.3 Optional additional questions

1. **Consistent estimators of the number of components.**

   (a) **Give a formal definition of consistent estimators of the number of components in a mixture model. Write some state-of-the-art on that topic, choose one of the references therein, justifying your choice. Provide a one-page description of the approach developed in that reference.**

   Estimation of number of components in mixture models is one of the hardest task, and thus there are many different approaches trying to suggest the most efficient and universal estimator. Like most of the problems in statistics we have two different main approaches: Frequentist and Bayesian, so we will take a look on both of them. Firstly, let see the definition of consistent estimator in the Frequentist approach.

   Definition 1. *Let $X_1, X_2, ..., X_n$ be i.i.d. random variables from a mixture model with K components. Then we say that the estimation of the number of clusters K_n is a consistent estimator for K if*

   $$(\forall \varepsilon > 0) \lim_{n\to\infty} P(|K_n - K| > \varepsilon) = 0.$$

   On the other hand, when it comes to Bayesian approach, we know that our parameter is now a random variable, so we need to speak of posterior consistency. But first, let us give some overview of this approach. We consider a Bayesian model

consisting of a prior distribution Π_0 on a parameter space Θ, and conditionally i.i.d. data governed by likelihood distribution P_θ, i.e.

$$\Theta \sim \Pi_0 \qquad X_1, X_2, ..., X_n \mid \Theta \sim P_\theta.$$

The posterior distribution Π on Θ after observing n observations $X_1, X_2, ..., X_n$ is defined by

$$(\forall A \subseteq \Theta)\ \Pi(A \mid X_1, X_2, ..., X_n) = \frac{\int_A \prod_{i=1}^{n} p_\theta(X_i) d\Pi_0(\theta)}{\int_\Theta \prod_{i=1}^{n} p_\theta(X_i) d\Pi_0(\theta)}.$$

Here, of course we presumed that A is measurable. When it comes to mixture model with an unknown number of components, the likelihood P_θ is a weighted sum of component distributions F_ξ, i.e.

$$(\forall A \subseteq \mathcal{X})\ P_\theta(A) = \sum_{k=1}^{K} \pi_k F_{\xi_k}(A),$$

where each parameter $\theta \in \Theta$ can be expressed as

$$\theta = (k, \pi_1, \pi_2, ..., \pi_k, \xi_1, \xi_2, ..., \xi_k),$$

for some $k \in \mathbb{N}$.

Definition 2. *Suppose our model is well-specified in that the data $X_1, X_2, ..., X_n$ are truly generated from P_θ for some $K \in \mathbb{N}$. We say that posterior distribution of number of components K is consistent if*

$$(\forall k \in \mathbb{N})\ \lim_{n\to\infty} \Pi(k \mid X_1, X_2, ..., X_n) = \mathbb{I}(k = K)\ \textit{almost surely.}$$

Of course, this last definition is not really strict because we need to know in respect to what probability the convergence is almost surely. But either way it gives us a good intuition what is happening in Bayesian world. According to [1]: only Bayesian can truly estimate K, that is, treat K as an additional unknown parameter that can be estimated simultaneously with the other model parameters

$$\theta = (\pi_1, \pi_2, ..., \pi_K, \xi_1, \xi_2, ..., \xi_K)$$

defining the mixture distribution.

Now, when we saw definitions, let's take a look at what are some of the techniques to find the consistent estimator. As far as Frequentist approach concerns there are several consistent estimators under some assumptions. One of them is proposed by Keribin in [2], it is based on maximum penalized likelihood estimation,

for appropriate penalization sequences. In order to prove consistency she made a lot of theoretical assumptions (such as f_ξ possesses partial derivatives until order five). Then, she showed that her estimator is consistent with Gaussian mixture model and Poisson mixture model. Also, this result can be applied just on one dimensional model, but it can be applied to a special multivariate normal mixture model under the assumption that a superior value for K is known. On the other hand Henna [3] suggested estimation for multivariate mixture model. He transformed data $\mathbf{X}$ to $\mathbf{Y}$ such that

$$\mathbf{Y}_\xi = \mathbf{M}\mathbf{X}_\xi + \rho,$$

where $\mathbf{M}$ is orthogonal matrix, and ρ is a column vector. So, he adopted a real valued function T_j such that

$$T_j(\mathbf{X}) = \mathbf{a}_j\mathbf{X} + \rho_j,$$

where $\mathbf{a}_j$ is the j-th row of $\mathbf{M}$ and ρ_j is the j-th coordinate of ρ. Then putting $Y_{j\xi} = T(\mathbf{X}_\xi)$, for $\xi = 1, 2, ..., n$, he had $(Y_{j1}, Y_{j2}, ..., Y_{jn})$, one dimensional i.i.d. sample from a finite mixture model with K components. So, in order to estimate K, he suggested to first construct an estimator $\widehat{K}_{jn}$ of the number K_j of components on the bases of $(Y_{j1}, Y_{j2}, ..., Y_{jn})$. And the final estimator is given by

$$\widehat{K}_n = \max_{1\leqslant j\leqslant k} \widehat{K}_{jn},$$

where k is the dimension of pdf's of mixing components. It is shown that estimator is consistent when we are working with k-dimensional normal distributions. Peter Schlattmann [4] found the consistent estimator for the number of components in the homogeneous Poisson case. He applied the non-parametric bootstrap such that a mixture algorithm is applied B times to bootstrap samples obtained from the original sample with replacement. Then he proposed the mode of the bootstrap distribution to be the estimation of the number of components K and proved that it is consistent estimator. More recently, Huang, Peng and Zhang [5] continued work of Keribin with new penalized likelihood method for estimating number of components of finite Gaussian mixture models. Their idea is to eliminate all the components for which the mixing probabilities are shrunk to zero. When a component is eliminated, the objective function of proposed method changes continuously. We will come back to investigate this method more, but before that let us look at Bayesian approaches.

Nobile [6] demonstrates that finite mixtures exhibit posterior consistency assuming the model is well-specified, i.e.

$$\Pi_0\{\theta \in \Theta \mid (\exists i \neq j)\ \xi_i = \xi_j \text{ or } \pi_i = \pi_j\} = 0$$

and the class of densities $\{p_\theta \mid \theta \in \Theta\}$ is identifiable up to duplicate components and component reordering. Then, he proved

$$(\forall k \in \mathbb{N}) \lim_{n\to\infty} \Pi(k \mid X_1, X_2, ..., X_n) = \mathbb{I}(k = K)\ \Pi_0 \text{ almost surely.}$$

In 2001, Ishwaran, James and Sun [7] showed that in a well-specified setting, the posterior distribution of the number of components does not asymptotically underestimate the number of components when assuming a stronger identifiable condition, i.e. if there exists a countable sequence of sets $\{A_i\}_{i=1}^n$ for which $\theta \neq \theta'$ means $p_\theta(A_i) \neq p_{\theta'}(A_i)$. Their result does not cover the case when the number of components is larger then the true number. In real-world problems, data is usually misspecified. Mixture models are often matched with a non-parametric Bayesian prior in order to discover the number of components. Recent work by Miller and Harrison [8, 9, 10] demonstrated that such models are severely inconsistent for the number of components, i.e. the probability of the correct number of components being recovered decreases to zero as the amount of data increases. Here, a non-parametric Bayesian prior for mixture models gives a prior with support on the natural numbers for any number of data points. Alternative for this approach is to consider a prior on the number of components that does not vary with data size but still maintains support on all possible positive integer numbers of components. This is called finite mixture model. Later on, in 2017, Cai, Campbell and Broderick [11] showed that even in the finite mixture model, the posterior number of components typically concentrates strictly away in from the generating number of components, when number is finite. They also showed that the estimate of the number of components diverges to infinity almost surely, as the number of data grows, just as in the non-parametric case.

Let's now go back to the method for Gaussian mixture models by penalized likelihood. We saw different approaches and what did they solve or not. We saw that for now there is no universal consistent estimator for number of components for arbitrary mixture model. There are some which works fine with Gaussian mixture models, so we should try to understand what happens there in order to try to think of any improvement. Apart from that, another reason for choosing this method is because it uses modified EM algorithm and it is useful to see how it is done.

Let $\mathbf{X}$ be d-dimensional random variable from Gaussian mixture distribution, i.e.

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k).$$

For identifiability of the Gaussian mixture model, let K be the smallest integer such that $\pi_k > 0$ for $1 \leqslant k \leqslant K$, and $(\mu_a, \Sigma_a) \neq (\mu_b, \Sigma_b)$ for $1 \leqslant a \neq b \leqslant K$. We already know that log-likelihood function for EM algorithm is

$$Q\left(\theta, \theta^{(t)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{Z_i}^{(t)}(k)(\log \pi_k + \log \mathcal{N}(\mathbf{x}_i; \mu_k, \Sigma_k)).$$

The previous expression contains $\log \pi_k$, whose gradient grows very fast when π_k

is close to zero. Thus, the authors of the paper suggest to choose

$$\log \frac{\varepsilon + \pi}{\varepsilon} = \log(\varepsilon + \pi) - \log \varepsilon,$$

where ε is a very small positive number, let say 10^{-6}. Here $\log(\varepsilon + \pi) - \log \pi$ is a monotonically increasing function of π, and it is shrunk to zero when the mixing probability π goes to zero. So, new log-likelihood function looks like

$$Q_P\left(\theta, \theta^{(t)}\right) = Q\left(\theta, \theta^{(t)}\right) - n\lambda D_f \sum_{k=1}^{K} \left(\log(\varepsilon + \pi_k) - \log \varepsilon\right),$$

where λ is a tuning parameter, and D_f is the number of free parameters for each component. For Gaussian mixture model with arbitrary covariance matrices, each component has

$$D_f = 1 + d + \frac{d(d+1)}{2} = \frac{1}{2}d^2 + \frac{3}{2}d + 1$$

number of free parameters. We can notice that D_f is a constant and can be removed from the likelihood function, it simplifies the search range of λ in numerical study. In the E step, as before, we have current estimate

$$\theta^{(0)} = \left(\pi_1^{(0)}, \pi_2^{(0)}, ..., \pi_K^{(0)}, \mu_1^{(0)}, \mu_2^{(0)}, ..., \mu_K^{(0)}, \Sigma_1^{(0)}, \Sigma_2^{(0)}, ..., \Sigma_K^{(0)}\right),$$

and calculate the posterior probability $\gamma_{Z_i}^{(0)}(k)$. In the M step, on the other hand, we need to maximize the expected penalized log-likelihood function. Doing this firstly in respect to π_k and using Lagrange multipliers, we obtain the update rule

$$\pi_k^{(1)} = \max\left\{0, \frac{1}{1 - K\lambda D_f}\left(\frac{n_k^{(0)}}{n} - \lambda D_f\right)\right\}.$$

Some $\pi_k^{(1)}$ may be shrunk to zero, and, subsequently, the constraint $\sum_{k=1}^{K} \pi_k^{(1)} = 1$, may not be satisfied. However, this neither decreases the likelihood function, nor affects the estimate of the posterior probability $\gamma_{Z_i}^{(1)}(k)$ in the E step, or the update of pi_k in the M step. We only need to normalize $\pi^{(T)}$ by enforcing $\sum_{k=1}^{K} \pi_k^{(T)} = 1$ after the EM algorithm converges. The update equations of the rest of the parameters are the same as those of the standard EM algorithm for Gaussian mixture models, we already saw. This algorithm starts with a prespecified large number of components, and whenever a mixing probability is shrunk to zero, the corresponding component is deleted, and thus fewer components are retained for the remaining EM iterations. In the notation, we abuse the notation of the number of components K at the beginning of each EM iteration. Through the updating process, K becomes smaller and smaller. We need to modify this algorithm even

more, because the function $\log(\varepsilon + \pi_k) - \log \varepsilon$ would over penalize large π_k and yield a biased estimator. So, there is new penalized log-likelihood function

$$Q_P\left(\theta, \theta^{(t)}\right) = Q\left(\theta, \theta^{(t)}\right) - n\lambda D_f \sum_{k=1}^{K} \left(\log(\varepsilon + p_\lambda(\pi_k)) - \log \varepsilon\right),$$

where $p_\lambda(\pi)$ is the SCAD penalty function proposed by Fan and Li [12]. So, in this case the computation of update rules for π_k is lot more complex, and we are not going to cover it here. For the more details on this topic see [5]. To obtain the final estimate of the mixture model by maximizing any of Q_P, one needs to select the tuning parameters λ. There are many methods to do this, such as generalized cross-validation or Bayesian information criterion or BIC. We define a BIC

$$BIC(\lambda) = \sum_{i=1}^{n} \log\left(\sum_{k=1}^{\widehat{K}} \widehat{\pi}_k \cdot \mathcal{N}(\mathbf{x}; \widehat{\mu}_k, \widehat{\Sigma}_k)\right) - \frac{1}{2}\widehat{K} D_f \log n,$$

where $\widehat{K}$ is the estimate of the number of components and $\widehat{\pi}_k$, $\widehat{\mu}_k$ and $\widehat{\Sigma}_k$ estimators of the π_k, μ_k and Σ_k, respectively, for a given λ. Then, we choose λ such that

$$\widehat{\lambda} = \arg\max_{\lambda} BIC(\lambda).$$

At the end, we can mention that under some mild assumptions of the real parameters of Gaussian mixture models Haung, Peng and Zhang proved that the estimator for the number of components is consistent.

(b) **Imagine, describe and implement a protocol to evaluate the consistency of any arbitrary estimator of the number of components. Test this protocol on Gaussian mixtures to check the consistency of that estimator.**

Here, we want to test if the estimator is consistent in Frequentist point of view. For this of course, we need to know a true value for the number of components in order to see if the estimator is consistent. So, we need to check if the estimator will converge to the true value with probability 1. The main idea will be to simulate data from a mixture distribution with K components and apply estimator on this data. Then, we need to see how the estimated value is close to K. If the estimator is consistent, with increasing the sample size, the estimator should be more closer to the true value. At the end, we did Monte Carlo simulation to check if the probability of the estimator converging to the true value is close to one. We tried our function on the estimation of the components based on lowest BIC. By putting different number of components, we saw that this estimator is bad for small number of components. So, we can conclude that this estimator is not consistent.

2. **Implementation of the mixtures of von Mises distributions**

(a) **Write your own sampling function and pdf function of Mixtures of von Mises distributions.**
For this problem, we used function `numpy.random.vonmises` for generating random number from von Mises distribution. Then, like in the second question of the preparatory work we mixed the data, by random sampling for Z with appropriate probabilities.

(b) **Use your functions to simulate a 3-components mixture, with sample size of 1,000. Provide the figure showing the data colored by the true labels and the contour plot of the log(pdf) of the simulated model (you may visualize them on 2D euclidean space).**
We simulated data of size 1000 from mixture of von Mises distribution with next parameters

$$\theta = (\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3, \kappa_1, \kappa_2, \kappa_3) = (0.2, 0.5, 0.3, -2, 0, 2, 25, 60, 30).$$

We plotted a histogram of data and the pdf in the same plot. Also, we transformed data from angular to 2D Euclidean with cos and sin and plotted on the circle.

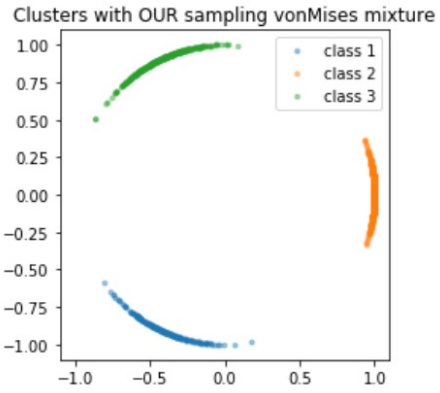


Figure 11: The sampling von Mises mixture

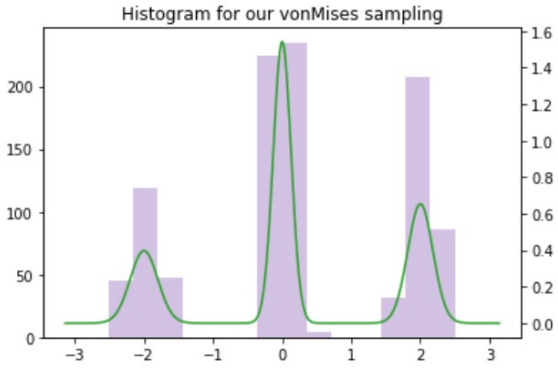


Figure 12: Histogram of the sampling von Mises mixture

We see on Figure 11 and 12 the 3 distinct component of the von Mises distribution. Now, we want to plot contour plot over 2D representation of our angular data. First of all, we are using -log(pdf) to get more sparse values. Secondly, we need to notice that our pdf is the function of angle, which means that for the points in 2D which lay on the same angle, but different radius we will have the same value of pdf. This will mean that our contours will not look like the one we had in GMM case. To show this, let us first plot surface of the -log(pdf) over data represented in Eucldean space (Figure 13). So, for every point in 2D we found the angle and we computed pdf of von Mises with given parameters in that angle.

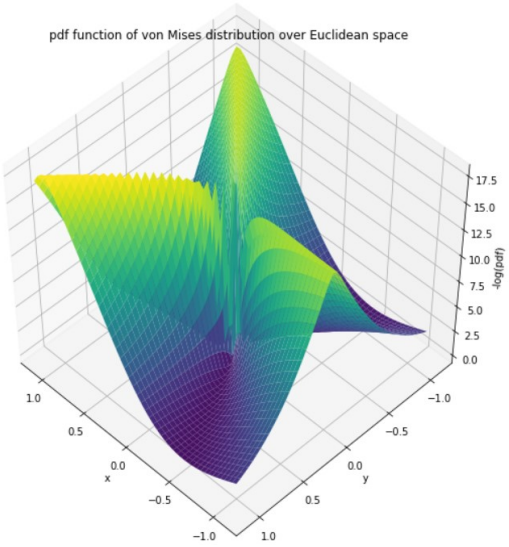


Figure 13: pdf of von Mises over 2D representation

Let us now look at the contours, which is basically the projection of the surface above to the 2D space (Figure 14).

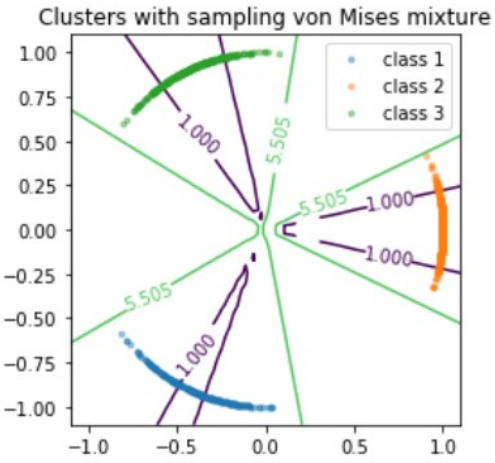


Figure 14: Contour plot of the log(pdf)

(c) **Estimate the parameters on the simulated data using your implementation. Comment the results using parameters, histograms and bivariate plots with clusters (the same plot as for (b) but using the estimated parameters).**

We implemented the EM algorithm based on the forth problem in the mandatory part and the results are

$$\begin{aligned}\widehat{\theta} &= (\widehat{\pi}_1, \widehat{\pi}_2, \widehat{\pi}_3, \widehat{\mu}_1, \widehat{\mu}_2, \widehat{\mu}_3, \widehat{\kappa}_1, \widehat{\kappa}_2, \widehat{\kappa}_3) \\ &= (0.187, 0.497, 0.316, -2.003, 0.006, 2.005, 27.945, 56.589, 29.352).\end{aligned}$$

We can see that the estimated parameters are close to the true values, so we can conclude that our EM algorithm works. Below, there is a histogram with two density lines, one with the true parameters and one with the estimated. We can see that those two lines are almost identical.

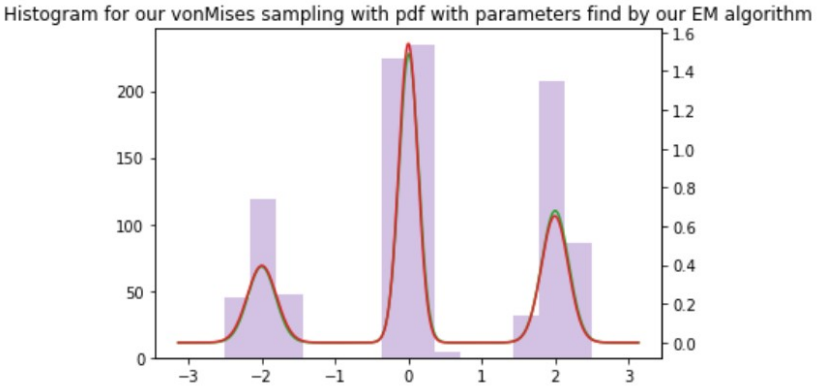


Figure 15: pdf with generated (in green) and estimated (in red) parameters

And, finally, if we would like to plot the contour plot from the previous question, but with estimated parameters, we will see in Figure 16 that there is no difference.

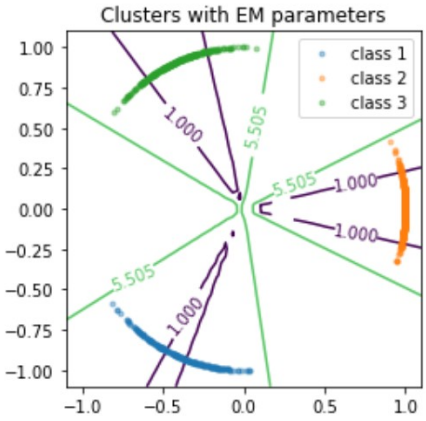


Figure 16: Contour plot of the log(pdf) with EM parameters

References

[1] Christian Robert Gilles Celeux, Sylvia Frühwirth-Schnatter. *Model Selection for Mixture Models-Perspectives and Strategies*. 2018.

[2] C. Keribin. Consistent estimation of the order of mixture models. *Sankhya: The Indian Journal of Statistics*, 62, 2000.

[3] Jogi Henna. Estimation of the number of components of finite mixtures of multivariate distributions. *Ann. Inst. Statist. Math.*, 57, 2005.

[4] Peter Schlattmann. Estimating the number of components in a finite mixture model: the special case of homogeneity. *Computational Statistics & Data Analysis*, 41, 2003.

[5] Heng Peng Tao Huang and Kun Zhang. Model selection for gaussian mixture models. *Statistica Sinica*, 27, 2017.

[6] A. Nobile. Bayesian analysis of finite mixture distributions. *PhD thesis, Carnegie Mellon University*, 1994.

[7] L. F. James H. Ishwaran and J. Sun. Bayesian model selection in finite mixtures by marginal density decompositions. *Journal of the American Statistical Association*, 2001.

[8] J. W. Miller and M. T. Harrison. A simple example of dirichlet process mixture inconsistency for the number of components. *Advances in Neural Information Processing Systems*, 2013.

[9] J. W. Miller and M. T. Harrison. Inconsistency of pitman-yor process mixtures for the number of components. *Advances in Neural Information Processing Systems*, 2014.

[10] J. W. Miller and M. T. Harrison. Mixture models with a prior on the number of components. *Advances in Neural Information Processing Systems*, 2016.

[11] Tamara Broderick Diana Cai, Trevor Campbell. Finite mixture models are typically inconsistent for the number of components. *31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[12] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Associatio*, 96, 2001.