Fundamentals of Probabilistic Data Mining, Graded lab.

CERQUEIRA PONTE, Joel
joelcponte@gmail.com

HUARTE SALAZAR, Ricardo
Ricardo.Huarte-Salazar@grenoble-inp.org

SU, Aimin
aiminsuhdu@gmail.com

LUU, Duc-Anh
duc-anh.luu@grenoble-inp.org

January 19, 2018

1 Introduction

In this report we present the Lab sessions results for the "Probabilistic Data Ming" Course along with the research-like assignments.

2 Probabilistic graphical models

2.1 Lab work

2.1.1 Simulated data

Firstly, simulate a Gaussian model with the perfect map in Figure 1. To do this, use linear regression models using offsets 0, the coefficients in the figure, and the residual standard deviation $\sigma = 1$.

1.Simulate one sample of size 40 and one sample of size 100. Compare the GS and

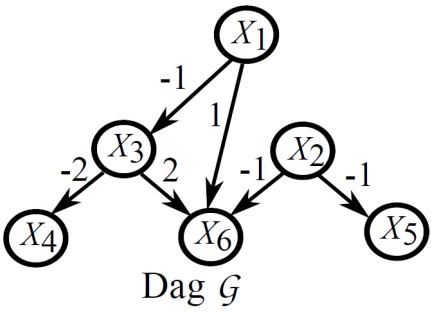


Figure 1: Simulated Model

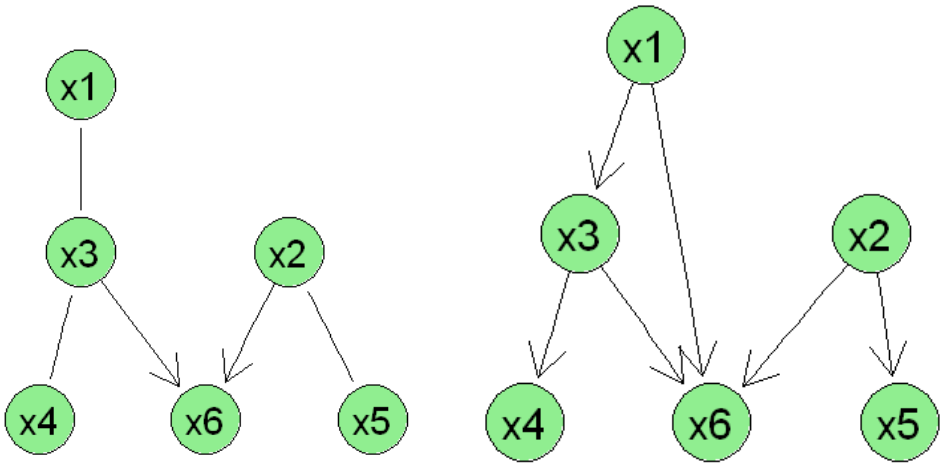
HC procedures (Scutari, 2010) using both sample sizes. What is your conclusion?

We can immediately see in Figure 2 that with a sample size of 40 GS is not able to find the correct model, as a result of the small size of the sample, however we can see that HC actually found the correct map despite the small sample size.

In Figure 3 we can see that HC, as expected, was also able to find the correct with the sample size of 100, but GS still wan't able to obtain do it, although it got closer.

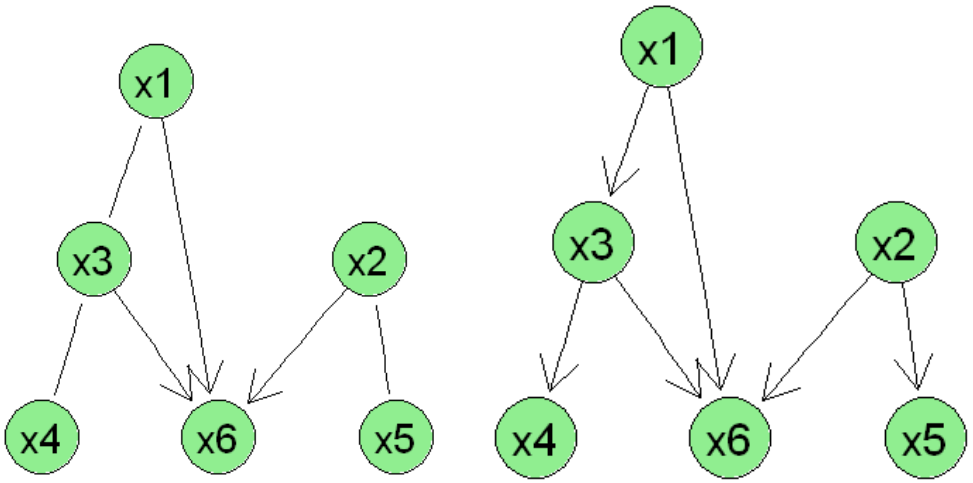
2.1.2 Real data: asset returns

We consider the returns of 8 assets on $n = 5039$ days. The daily return $X_{t,i}$ of asset i at time t is defined as $(V_{t,i}V_{t-1,i}) = V_{t,i}$, where $V_{t,i}$ is the value of asset i at time t.



(a) (b)

Figure 2: Size 40: (a)Grow-Shrink model. (b)Hill-Climbing model.



(a) (b)

Figure 3: Size 100: (a)Grow-Shrink model. (b)Hill-Climbing model.

Here, we consider only the assets: **"AIR.FRANCE.KLM", "ALCATEL.LUCENT" "AXA", "FAURECIA", "GAUMONT", "GEODIS", "PPR"** and **"UNION.FINC.FRANC."**

1. **Use file "Returns250d" to create a data frame with only the 8 assets listed above.**
   Please see the code in the attached Jupyter notebook.
2. **Estimate directed graphs using the gs and hc procedures (Scutari, 2010) and plot their graphs.** In figure 4 and 5 we can see the models obtained by **GS** and **HC**
3. **Find a marginal independence relationship between two variables found by gs but not by hc. Use ci.test to perform a statistical test of independence. What do you conclude?**

- *Marginal independence between GEODIS and ALCATEL.LUCENT*

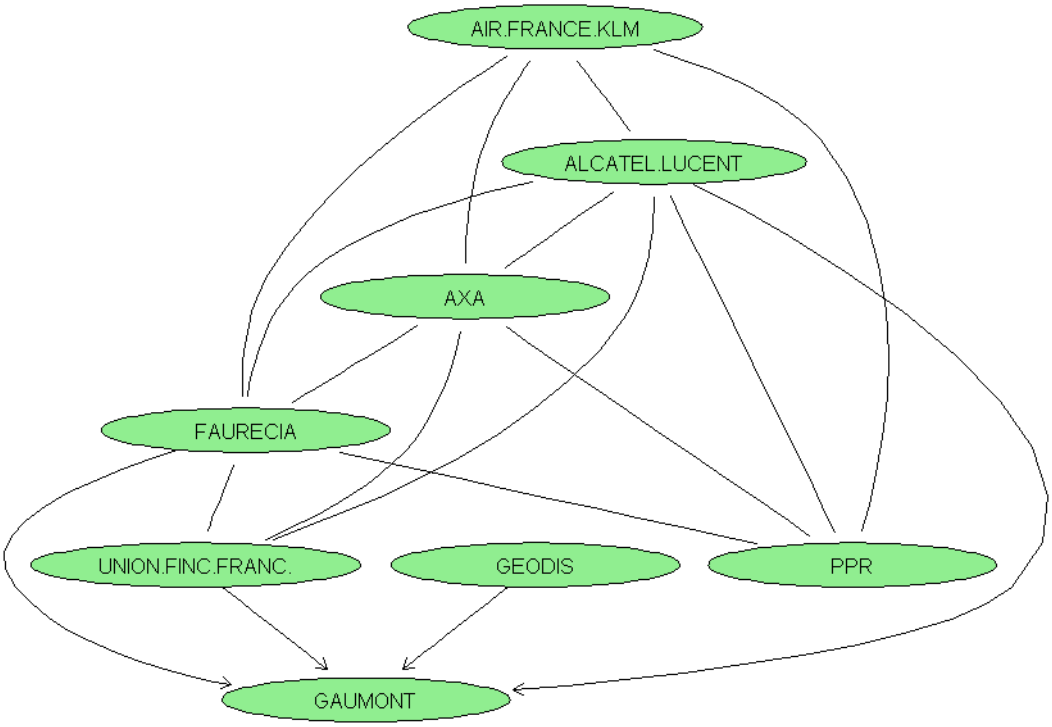


Figure 4: GS model for Asset returns

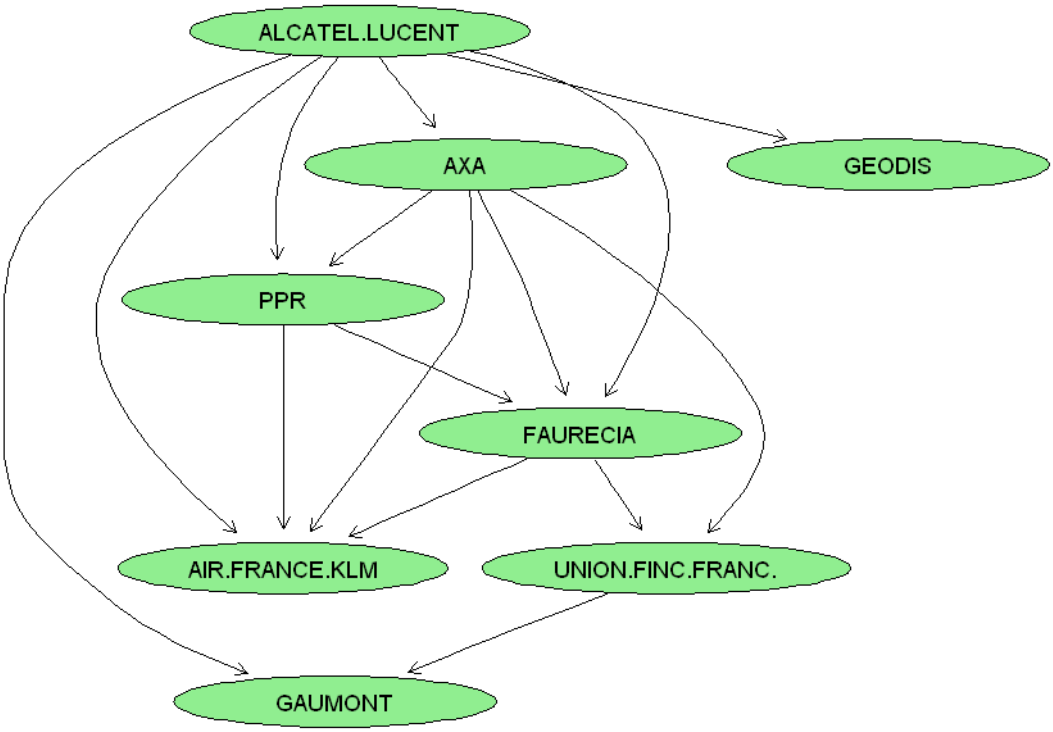


Figure 5: HC model for Asset returns

All the paths between GEODIS and ALCATEL pass through a "v-structure" pattern in the graph implied by GS, which indicated marginal independence. In the graph implied by HC they are directly connected, which means that they are not marginally independent according to it. Considering a 95% confidence level, the small p-value of $4.625 * 10^{-5}$ given by the CI

test indicates evidence against the null hypothesis, meaning that they are likely to not be marginally independent.

4. **Find a conditional independence relationship between two variables given another set of variables found by hc but not by gs. Use ci.test to perform a statistical test of (conditional) independence. What do you conclude?**

   - *Conditional independence between GEODIS and UNION.FINC.FRANC. given GAUMONT and ALCATEL.LUCENT*

   In the graph implied by HC, all the paths between GEODIS and UNION.FINC.FRANC. are blocked by ALCATEL.LUCENT. This implies that GEODIS AND UNION.FINC.FRANC. are conditionally independent given ALCATEL.LUCENT, which continues to be true if we include GAUMONT. For GS, we see a v-structure pattern "GEODIS->GAUMONT<-UNION.FINC.FRANC." which imply that they the conditional independence doesn't hold here. The same is true when we observe ALCATEL.LUCENT, additionally.

   Considering a 95% confidence level, the p-value of 0.1098 found by the CI test prevents us from rejecting the null hypothesis, meaning that they are likely to be conditionally independent.

5. **Find a conditional independence relationship between two variables given another set of variables found by both hc and gs. Use ci.test to perform a statistical test of (conditional) independence. What do you conclude?**

   - *GEODIS and UNION.FINC.FRANC. are conditionally independent given ALCATEL.LUCENT*

   For the graph generated by GS, we again have the v-structure "GEODIS->GAUMONT<-UNION.FINC.FRANC", which implies the conditional independence between GEODIS AND UNION.FINC.FRANC given ALCATEL.LUCENT.

   In HC, we again see that all paths between GEODIS and UNION.FINC.FRANC. are blocked by ALCATEL.LUCENT, which implies the same conditional independence.

   Considering a 95% confidence level, the p-value of 0.08058 found by the CI test prevents us from rejecting the null hypothesis, meaning that they are likely to be conditionally independent.

2.2 Mandatory additional questions

1. **We address the issue of consistent directed PGM estimation. Give a formal definition of consistent directed PGM estimation. Write some tate-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.**

   Consistency in PGM means that the estimator will obtain the true structure G or any structure G^* that is I-equivalent to the true structure G when the sample size $n \to \infty$.

   The PC Algorithm [1] for PGM estimation and its consistency is discussed in [2] where the uniform consistency of the algorithm is proven. It is also shown that PC consistently estimates the PGM with p being the number or variables and when the size $n \to \infty$, even when $p = p_n = O(n^a)(0 \leq a < \infty)$ grows as a function of n. PC algorithm starts from an undirected graph and based on zero order conditional independence recursively eliminates edges, as first the PC algorithm seeks to find the skeleton and later to find the Complete Partially Directed Acyclic Graph (CPDAG).

   Being the data, realizations of i.i.d. random vectors $X_1, ..., X_n$ with $X_i \in \mathbb{R}^p$ from a DAG G with distribution P and letting the dimension grow as a function of the size, $p = p_n$ and DAG $G = G_n$ and the distribution $P = P_n$, the assumptions needed for consistency as stated in [2] are:

   (a) For all n, P_n is multivariate Gaussian and faithful to the G_n.

(b) The dimension $p_n = O(n^a)$ for some $0 \leq a < \infty$.

(c) The maximum number of neighbors in G_n is denoted by $q_n = \max_{1 \leq j \leq p_n} |adj(G, j)|$, with $q_n = O(n^{1-b})$ for some $0 < b \leq 1$.

(d) The partial correlations between X^i and X^j given $\{X^r; r \in k\}$ for some set $k \subset \{1, ..., p_n\}$, $\{i, k\}$ are denoted by $\rho_{n;,i,j|k}$. Their absolute values are bounded by:

$$\inf\{|\rho_{i,j|k}|; i, j, k \text{ with } \rho_{i,j|k} \neq 0\} \geq c_n, c_n^{-1} = O(n^d),$$

$$\text{for some } 0 < d < \frac{b}{2} \text{ where } 0 < b \leq 1 \text{ is like in Assumbtion } b$$

$$\sup_{\rho_{n;,i,j|k}} |\rho_{i,j|k}| \leq M < 1$$

In a high-level, the algorithm is described as follows: The first part is to find the skeleton using the population version of the PC-algorithm 1 which assumes perfect knowledge of all conditional independence relationships

Algorithm 1 The PC_{pop}-algorithm

Input: Vertex Set V and the Conditional Independence Information
Output: Estimated Skeleton C and separation sets S

From the undirected graph $\tilde{C}$ on V
$\ell = -1 = \ell + 1$
repeat $\ell = \ell + 1$
  repeat Select new ordered pair of nodes i, j that are adjacent in C s.t. $|adj(C, i) \backslash \{j\}| \geq \ell$
    repeat Choose new $\mathbf{k} \subseteq adj(C, I) \backslash \{j\}$ with $|\mathbf{k}| = \ell$
      if i and j are conditionally independent given **k then**
        Delete edge i, j
        Denote this as new graph by C
        Save **k** in $S(i, j)$ and $S(j, i)$
      end if
    until edge i,j is deleted for all $\mathbf{k} \subseteq adj(C, I) \backslash \{j\}$ with $|\mathbf{k}| = \ell$ were chosen
  until all ordered pairs of adjacent variables i and j, s.t. $|adj(C, i) \backslash \{j\}| \geq \ell$ and $\mathbf{k} \subseteq adj(C, I) \backslash \{j\}$ with $|\mathbf{k}| = \ell$ were tested for **CI**
until for each ordered pair of adjacent nodes $i, j : |adj(C, i) \backslash \{j\}| < \ell$

The second part of the algorithm 2 is aimed to find the CPDAG from the Skeleton

Algorithm 2 From the Skeleton to a CPDAG

Input: Skeleton G_skel, separation sets S
Output: CPDAG G

for all pairs of nonadjacent variables i,j with common neighbor k **do**
  if $k \in S(i, j)$ **then**
    Replace x
  end if
end for
In the resulting PDAG, attempt orienting as many undirected edges as plausible by applying the following rules:
R1 Orient $j - k$ into $j \to k$ whenever there is an arrow $i \to j$ such that i and k are nonadjacent.
R2 Orient $i - j$ into $i \to j$ whenever there is a chain $i \to k \to j$.
R3 Orient $i - j$ into $i \to j$ whenever there are two chains $i - k \to j$ and $i - l \to j$ such that k and l are nonadjacent.
R4 Orient $i - j$ into $i \to j$ whenever there are two chains $i - k \to l$ and $k \to l \to j$ such that k and l are nonadjacent.

2. **Apply the chosen approach to the asset returns data set. Compare it with the results of hc in part 2.1.2. Use ci.test to try to evaluate the relevance of proposals**

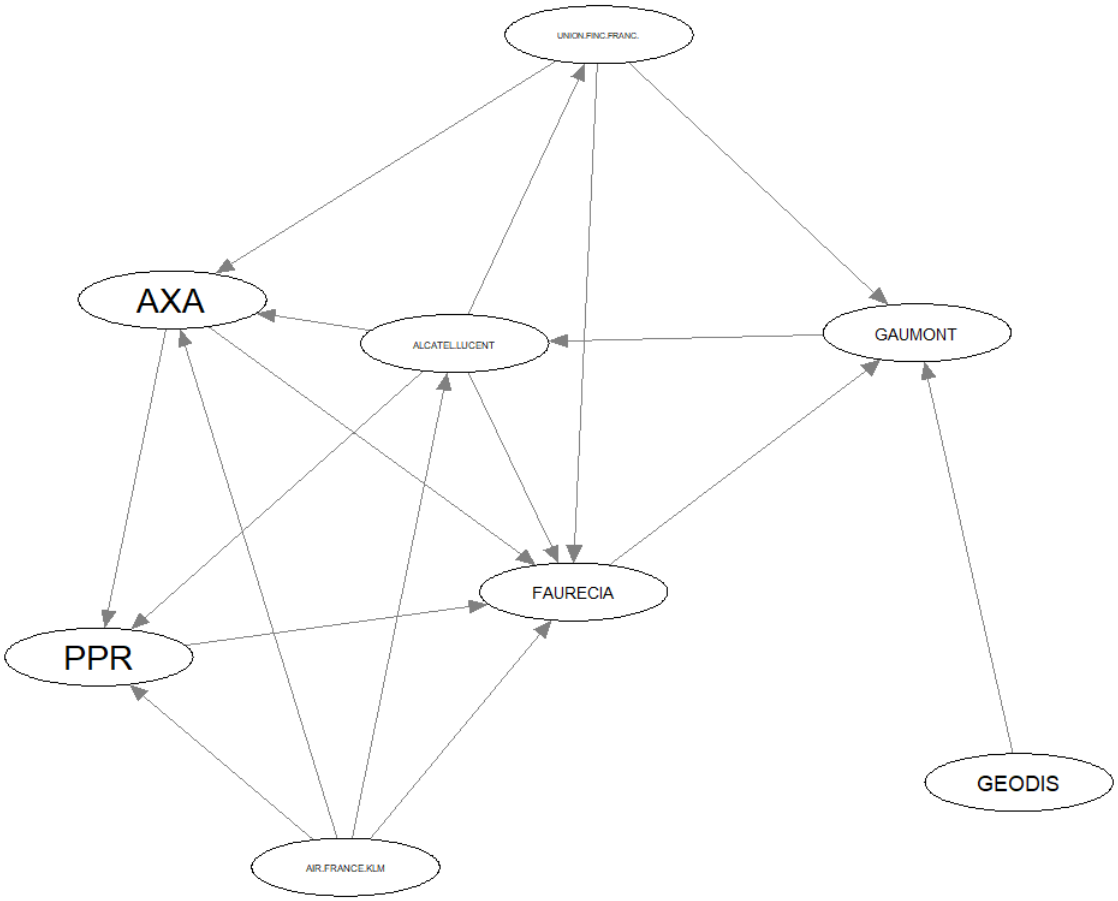


Figure 6: Implementation of PC on the Asset returns dataset.

for edges where both methods yield different results.

For **PC** we used the package pcalg in R for structure learning and parameter estimation of Bayesian networks. The code for this R package is available in (CRAN) at https://CRAN.R-project.org/package=pcalg.

We evaluated conditional independence tests for two CI implied by HC but not PC: ALCATEL.LUCENT and UNION.FINC.FRANC. given AXA and PPR; and GEODIS and AXA given ALCATEL.LUCENT.

They yielded p-values of 0.0042 and 0.031, giving evidence that PC is more correct.

2.3 Optional additional questions

1. **Imagine, describe and implement a protocol to evaluate consistency of any arbitrary directed PGM estimation method. Test this protocol on the method chosen in part 2.2 and provide the result. What is your conclusion? What is the effect of the number of variables?**

   We aim to check if a method yields the correct (or I-equivalent) graph under infinite data. To do that, we can use simulated data and study the asymptotic behavior of the method by learning from samples of increasing size.

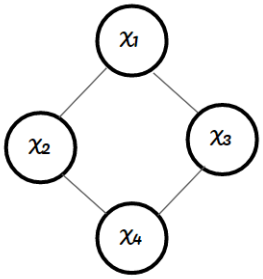


Figure 7

For each generated sample we will know the true graph. So, we can check a consistent score such as BIC for the true graph and for the generated graph. As the sample size increases, if we use a consistent method, we expect these two scores to either be the same (if we let the sample size be big enough) or get asymptotically closer. We should repeat this process with a increasing number of variables as well. The expected behavior is that, the more variables you have, the bigger the samples will need to be in order to reach the correct graph.

2. **What are the assumption of the method chosen in part 2.2? Check as many assumptions as you can on the asset returns data set.**

   The assumptions of the PC algorithm are described in section 2.2.1, Assumption b is complied as in the case of the returns dataset the number of variables p does not grow as a function of n. Assumption 3 is a sparseness assumption which holds true also when p is fixed

3. **Define and simulate some 4-variable model that cannot have a perfect directed map. Estimate a directed PGM using the method chosen in part 2.2. What do you obtain? Why? (How to interpret this result?)**

   We define the model shown in figure 7 (see Jupyter notebook for details) which cannot have a perfect directed map, as there is no direct map that perfectly captures the independences in the distribution.

   The learned graph is presented in figure 8. We can see that it was able to find the correct undirected graph. That can be explained from the fact that PC first tries to retrieve the skeleton, and later the directions.

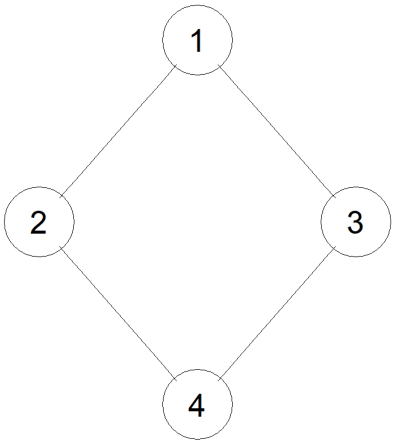


Figure 8: Learned graph.

4. **If two methods claim to be consistent but yield different DAGs on a same data set, how would you used both graph to propose a new graph? What would the expected properties of that graph be?**

   Say G_1 is a graph estimated from method M_1 and G_2 the graph estimated from M_2. One way could be to use the learning method G_2 starting with the graph M_1, resulting in the

graph G_{21} and vice-versa, resulting in the graph G_{12}. Then, to get the graph that yields the best score, given a consistent score function.

3 Independent mixture models

3.1 Lab work

3.1.1 Modeling

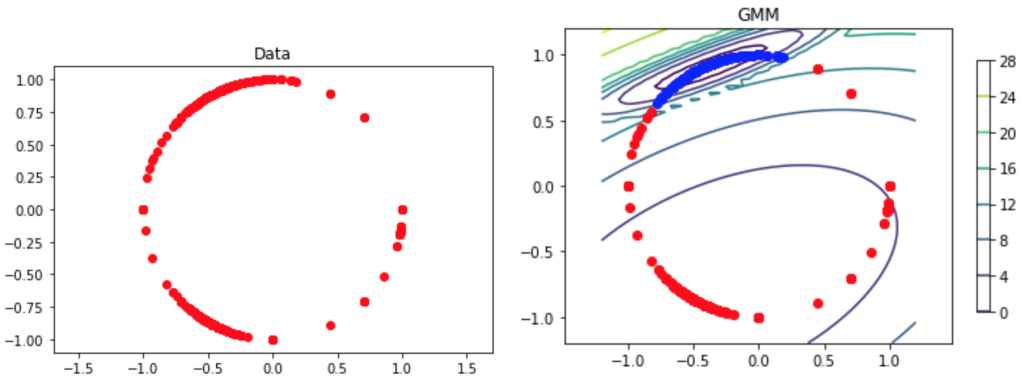
A priori (before reading the data), do you think a two-state Gaussian model could be appropriate? Why?

It seems that Gaussian mixtures in this case works. Because the stroke vectors associated with letter A have 2 main directions so, intuitively, these points will form 2 clusters.

3.1.2 Data analysis: Gaussian model

1. **Plot the data set. Estimate model and prove the re-estimation formula (exercise 3.3 in the booklet)**

You can see the plot of dataset in figure 9a.



(a) (b)

Figure 9: (a)Plot of the dataset. (b)Labeled plot of the dataset with GMM.

By running the estimating procedure several times, we get the model having highest score (log-likelihood) with parameters:

- Weights

$$\begin{bmatrix} 0.52639398 & 0.47360602 \end{bmatrix}$$

- Means

$$\begin{bmatrix} \begin{bmatrix} -0.29510373 & -0.72483125 \end{bmatrix} \\ \begin{bmatrix} -0.36571287 & 0.9060224 \end{bmatrix} \end{bmatrix}$$

- Covariances

$$\begin{bmatrix} 0.27188755 & 0.0823439 \\ 0.08234390 & 0.1156479 \end{bmatrix}$$
$$\begin{bmatrix} 0.03903012 & 0.0142656 \\ 0.01426564 & 0.0063493 \end{bmatrix}$$

EM algorithm for mixture of Gaussians:
Repeat until convergence: {
(E-Step): For each training example i, set:

$$\gamma_{ik} = p(z = k|x_i) = \frac{p(x_i|z = k)p(z = k)}{\sum_{k=1}^{K} p(x_i|z = k)p(z = k)} = \frac{\pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)}$$

(M-Step): Update the parameter

$$\begin{aligned}
\pi_k &= \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik} \\
\mu_k &= \frac{\sum_{i=1}^{N} \gamma_{ik} x_i}{\sum_{i=1}^{N} \gamma_{ik}} \\
\Sigma_k &= \frac{\sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^\top}{\sum_{i=1}^{N} \gamma_{ik}}
\end{aligned}$$

}

Proof of re-estimation formulas: Given data set of N points $x_i, i = 1, ..., N$, find mixture of Gaussians (MoG) that best explains data. We assume that data are drawn independently from MoG. Now we want to maximize data log-likelihood w.r.t parameters of MoG:

$$\lambda = \arg\max_{\lambda \in \mathcal{C}} \sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k p_{\theta_k}(x_i) = \arg\max_{\lambda \in \mathcal{C}} l_{x_1,...,x_N}(\lambda)$$

With $\mathcal{C} = \{(\pi_1, ..., \pi_K, \theta_1, ..., \theta_K) | \sum_k \pi_k = 1 \quad \text{and} \quad \forall k, \pi_k \geq 0\}$. Assuming that $\gamma_{ik} = p(z = k|x_i)$, so $\sum_k \gamma_{ik} = 1$. Considering Jensen's inequality for concave function: $\mathbb{E}[f(X)] \leq f(\mathbb{E}[X]$ with $f(x) = \ln(x)$, we have

$$\sum_{i=1}^{N} \ln \sum_{k=1}^{K} \gamma_{ik} \frac{\pi_k p_{\theta_k}(x_i)}{\gamma_{ik}} \geq \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \ln \frac{\pi_k p_{\theta_k}(x_i)}{\gamma_{ik}}$$

Now, to find parameters of mixture of Gaussians model, we need to maximize the quantity

$$\begin{aligned}
&\sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \ln \frac{\frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp(-\frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k)) \cdot \pi_k}{\gamma_{ik}} \\
&\text{s.t} \qquad \forall k, \pi_k \geq 0, \sum_k \pi_k = 1
\end{aligned}$$

Building Langragian for this problem, we have

$$\mathcal{L}(\theta, \lambda) = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \ln \frac{\frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp(-\frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k)) \cdot \pi_k}{\gamma_{ik}} + \lambda(1 - \sum_{k=1}^{K} \pi_k)$$

Taking the derivative with respect to π_k and λ, then setting it to zero, we get:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\theta, \lambda)}{\partial \pi_k} = \frac{\sum_{i=1}^{N} \gamma_{ik}}{\pi_k} - \lambda = 0 \\
&\Rightarrow \pi_k = \frac{\sum_{i=1}^{N} \gamma_{ik}}{\lambda} \\
&\frac{\partial \mathcal{L}(\theta, \lambda)}{\partial \lambda} = 1 - \sum_{k=1}^{K} \pi_k = 0 \\
&\Leftrightarrow \sum_{k=1}^{N} \frac{\sum_{i=1}^{N} \gamma_{ik}}{\lambda} = 1 \\
&\Rightarrow \lambda = N \\
&\Rightarrow \pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}
\end{aligned}$$

Taking the derivative with respect to μ_l, we find

$$\begin{aligned}
&\nabla_{\mu_l} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \ln \frac{\frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp(-\frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k)) \cdot \pi_k}{\gamma_{ik}} \\
&= -\nabla_{\mu_l} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k) \\
&= \frac{1}{2} \sum_{i=1}^{N} \gamma_{il} \nabla_{\mu_l} 2\mu_l^\top \Sigma_l^{-1} x_i - \mu_l^\top \Sigma_l^{-1} \mu_l \\
&= \sum_{i=1}^{N} \gamma_{il} (\Sigma_l^{-1} x_i - \Sigma_l^{-1} \mu_l)
\end{aligned}$$

Setting this to zero and solving for μ_l, we find

$$\mu_l = \frac{\sum_{i=1}^{N} \gamma_{il} x_i}{\sum_{i=1}^{N} \gamma_{il}}$$

Taking the derivative with respect to Σ_l, we have

$$\begin{aligned}
&\nabla_{\Sigma_l} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \ln \frac{\frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp(-\frac{1}{2}(x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k)) \cdot \pi_k}{\gamma_{ik}} \\
&= -\nabla_{\Sigma_l} \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{ik} \frac{1}{2}((x_i - \mu_k)^\top \Sigma_k^{-1}(x_i - \mu_k) + \ln|\Sigma_k|) \\
&= \frac{1}{2} \sum_{i=1}^{N} \gamma_{il} (\Sigma_l^{-2}(x_i - \mu_l)^\top (x_i - \mu_l) - \Sigma_l^{-1})
\end{aligned}$$

Setting this to zero and solving for Σ_l, we find

$$\Sigma_l = \frac{\sum_{i=1}^{N} \gamma_{il}(x_i - \mu_l)(x_i - \mu_l)^\top}{\sum_{i=1}^{N} \gamma_{il}}$$

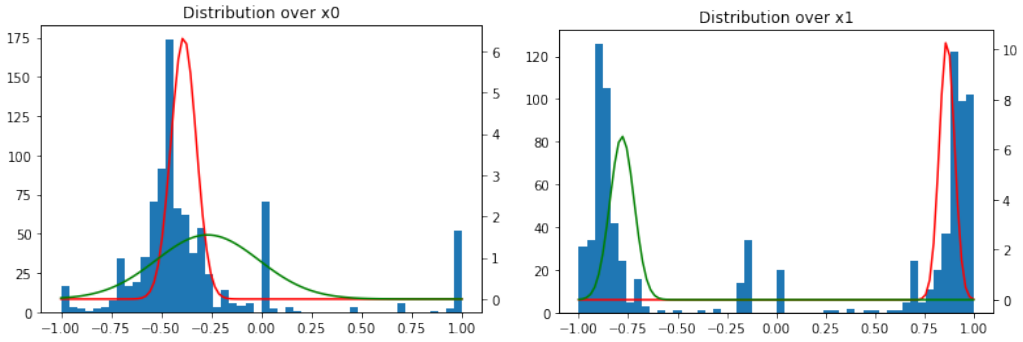
In the M-step, it updates the parameters of model based on observed $z = \{z_1, ..., z_k\}$. Since z is observed, the maximization of log-likelihood becomes easy to solve. Summary, we have:

$$\pi_k = \frac{1}{N}\sum_{i=1}^{N}\gamma_{ik}$$
$$\mu_k = \frac{\sum_{i=1}^{N}\gamma_{ik}x_i}{\sum_{i=1}^{N}\gamma_{ik}}$$
$$\Sigma_k = \frac{\sum_{i=1}^{N}\gamma_{ik}(x_i-\mu_k)(x_i-\mu_k)^\top}{\sum_{i=1}^{N}\gamma_{ik}}$$

2. **Label the data using the estimated model.**

Labeled plot for the dataset after applying the GMM can be found in figure 9b.

3. **Propose and implement a graphical (visual) method to validate the assumption of bivariate Gaussian emission distributions. What to think about this assumption?**



(a) (b)

Figure 10: (a)Distribution of first column data in the dataset. (b)Distribution of second column data in the dataset.

From figure 9b, we observe that the formed clusters by optimal model (which has the highest score) are not accurate. Now, we investigate the model more precisely.

We draw the distribution over each feature. We also show the estimated Gaussian density over each dimension in order to compare with these above distributions. The distribution of dataset is shown in figure 10. It depicts that there are two Gaussian mixtures in 10b while only one in 10a, which implies that bivariate Gaussian model may not be appropriate.

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

Von Mises distribution is a continuous probability distribution on the unit circle. It may be thought of as the circular analogue of the normal distribution. The Von Mises probability density function for the angle x given by:

$$p(x;\mu,\kappa) = \frac{e^{\kappa cos(x-\mu)}}{2\pi I_0(\kappa)}$$

Where μ is the mode and κ is the dispersion, and $I_0(\kappa)$ is the modified Bessel function of order 0. The parameters μ and $\frac{1}{\kappa}$ are analogous to μ and σ^2 (the mean and variance) in the normal distribution

Mixture of Von Mises distributions: Mixture density is weighted sum of Von Mises densities:

$$p(x) = \sum_{k=1}^{K} \pi_k \frac{e^{\kappa_k cos(x-\mu_k)}}{2\pi I_0(\kappa_k)}$$

with $\forall k, \pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$. So parameters are π_k, μ_k and κ_k

5. **A priori, would a mixture of Von Mises distributions be more or less adequate than Gaussian mixtures on the real data set of part 3.1? Why?**

Mixture of Von Mises seems to be more adequate than Gaussian Mixtures on this data set. Because intuitively, the points in the data set all belong to a circle in the 2D space, which is totally adapted to Von Mises distribution, when Gaussian Mixtures works well with data points spreading over all directions.

3.2 Mandatory additional questions

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

1. **Extend the scikit-learn mixture library by implementing mixtures of Von Mises distributions. Justify the E-step of the EM algorithm with equations.**

Justify E-step: starting from complete-likelihood with m training examples:

$$\begin{aligned}
\sum_{i=1}^{m} \log p(x_i; \theta) &= \sum_i \log \sum_k p(x_i, z = k; \theta) \\
&= \sum_i \log \sum_k \gamma_{ik} \frac{p(x_i, z = k; \theta)}{\gamma_{ik}} \\
&\geq \sum_i \sum_k \gamma_{ik} \log \frac{p(x_i, z = k; \theta)}{\gamma_{ik}}
\end{aligned}$$

The third line follows by using Jensen's inequality for concave function when γ_{ik} is a distribution over z. Until now, we have found the lower bound for log-likelihood. It is natural to try to make the lower-bound tight at that value of θ. To do that, we need the inequality above holds with equality. For this to be true, it is sufficient that the expectation taken over a constant. I.e we requires that:

$$\frac{p(x_i, z_i; \theta)}{\gamma_{ik}} = c$$

with c does not depend on z. And since γ_{ik} is a distribution over z ($\sum_k \gamma_{ik} = 1$), this tells us that:

$$\begin{aligned}
\gamma_{ik} &= \frac{p(x_i, z = k; \theta)}{\sum_k p(x_i, z = k; \theta)} \\
&= \frac{p(x_i, z_i; \theta)}{p(x_i; \theta)} \\
&= p(z = k | x_i)
\end{aligned}$$

EM algorithm of Mixtures of Von Mises distributions:
Repeat until convergence:{
(E-Step): For each training example i, set:

$$\gamma_{ik} = p(z = k|x_i) = \frac{p(x_i|z = k)p(z = k)}{\sum_{k=1}^{K} p(x_i|z = k)p(z = k)} = \frac{\pi_k \mathcal{V}(x_i; \mu_k, \kappa_k)}{\sum_{k=1}^{K} \pi_k \mathcal{V}(x_i; \mu_k, \kappa_k)}$$

(M-Step): Update the parameter

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}$$

$$\mu_k = \arctan(\frac{\sum_{i=1}^{N} \gamma_{ik} \sin x_i}{\sum_{i=1}^{N} \gamma_{ik} \cos x_i})$$

$$A(\kappa_k) = \frac{\sum_{i=1}^{N} \gamma_{ik} \cos(x_i - \mu_k)}{\sum_{i=1}^{N} \gamma_{ik}}$$

The value of κ_k can be computed by approximating the value of $A_2^{-1}(x)$ by

$$A_2^{-1}(x) \approx \frac{2x - x^3}{1 - x^2}$$

See VonMisesMixture.py for the detail implementation.

2. **What is a consistent estimator of mixture parameters? Imagine, describe and implement a protocol to evaluate consistency of any arbitrary estimator. Test this protocol on the algorithm developed in the previous question.**

Suppose θ is the mixture parameter. An estimator Tn of parameter θ is said to be consistent, if it converges in probability to the true value of the parameter.

$$p - \lim_{n \to \infty} T_n = \theta$$

But in many situations, parameter θ is actually unknown. And thus the convergence in probability must take place for every possible value of this parameter. Suppose that $p_\theta : \theta \in \Theta$ is a family of distributions, and $X^\theta = \{X_1, X_2, ...X_i \sim p_\theta\}$ is an infinite sample from the distribution p_θ. Let $T_n(X^\theta)$ be a sequence of estimators for some parameter $g(\theta)$. Usually T_n will be based on the first n observations of a sample. Then this sequence T_n is said to be consistent if,

$$p - \lim_{n \to \infty} T_n(X^\theta) = g(\theta), \text{for} \quad \theta \in \Theta$$

To evaluate consistency of the Von Mises Mixture parameters, we split the training data into different proportions. Then the model was trained with different scale of data, and we compare the parameters of the model. As is shown in 11, parameters of Von Mise mixture including means, weights and kappas are getting convergence by enlarging the size of training data. The x-axis of each sub-figure here is the proportion of the original training data. As the limit of the size of original dataset, some parameters haven't become convergence. But form the overall trend, we can expect that a better result will be obtained with larger dataset.

3. **Use a 2-state mixture of Von Mises distributions on the real data set of part 3.1. Transform the data to angular data. Provide a quantitative and a graphical way to compare the fitted mixture of Von Mises distributions with the Gaussian mixture obtained in part 3.1.**

In order to evaluate quantitatively the fitted mixture models, we use Bayesian information criterion (BIC) for the each model on the angular data input:

$$BIC\_GMM = 2428.52034$$
$$BIC\_VMM = 2163.12680$$

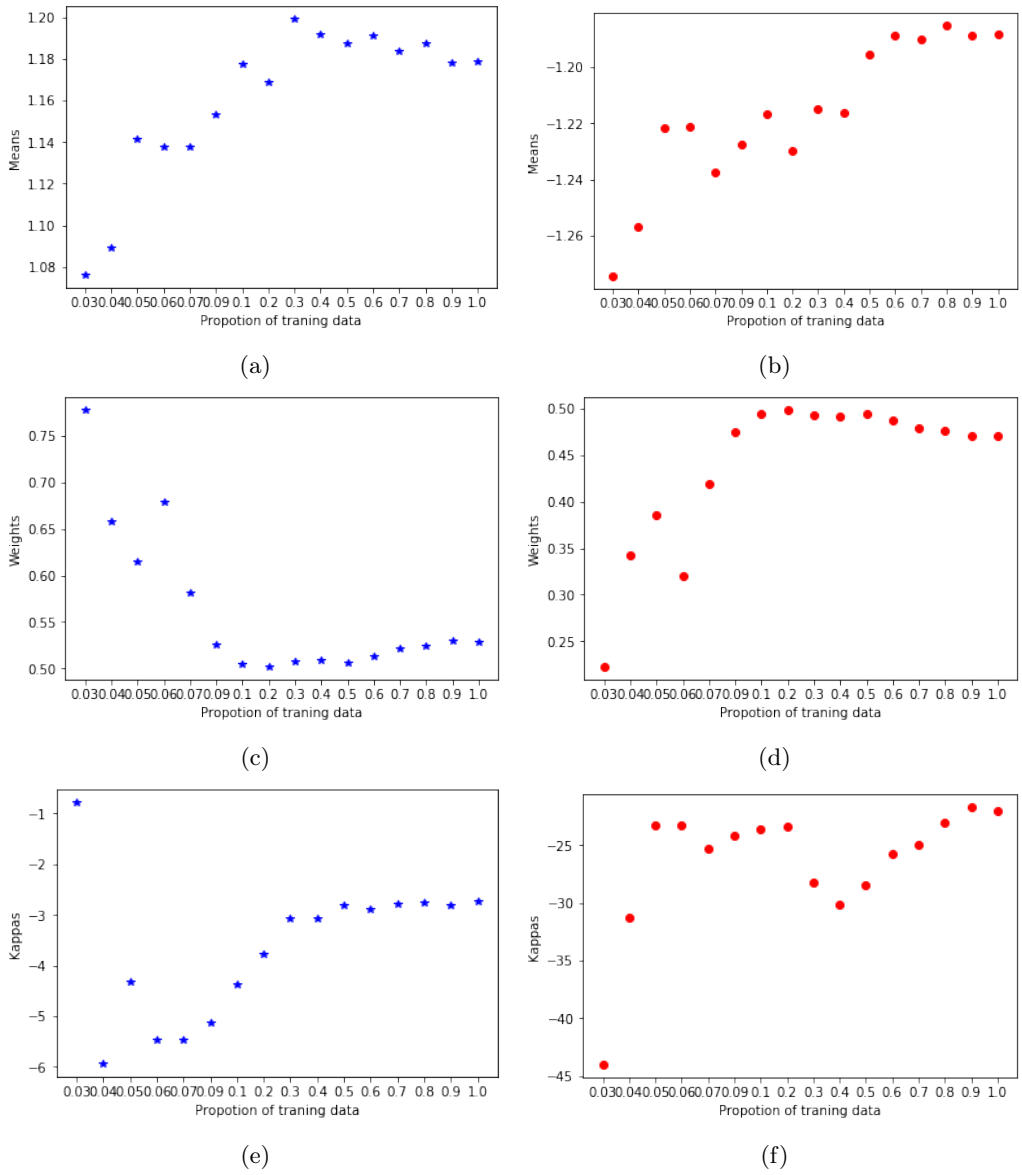


Figure 11: Parameters of Von Mixture are getting convergence by enlarging the training data size.

The lower BIC, the better model. So we see that Von Mises Mixtures model gives us better result in this case. The parameters of optimal model after running several times are:

- Weights

$$\begin{bmatrix} 0.52930556 & 0.47069444 \end{bmatrix}$$

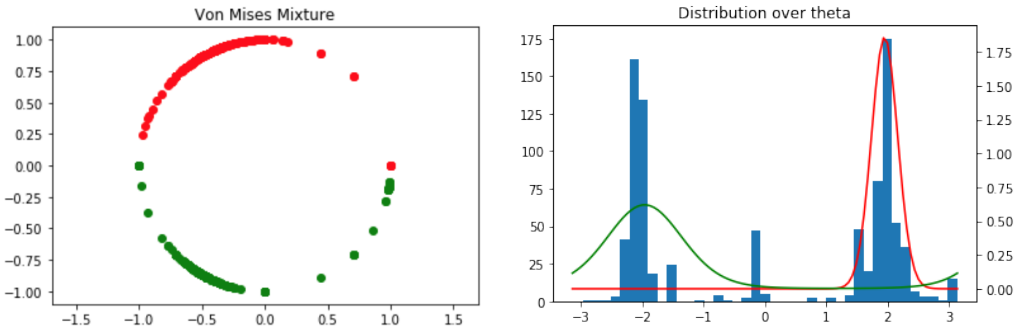
- Means

$$\begin{bmatrix} -1.96350436 & 1.9533927 \end{bmatrix}$$

- Kappas

$$\begin{bmatrix} 2.71306742 & 22.05319875 \end{bmatrix}$$

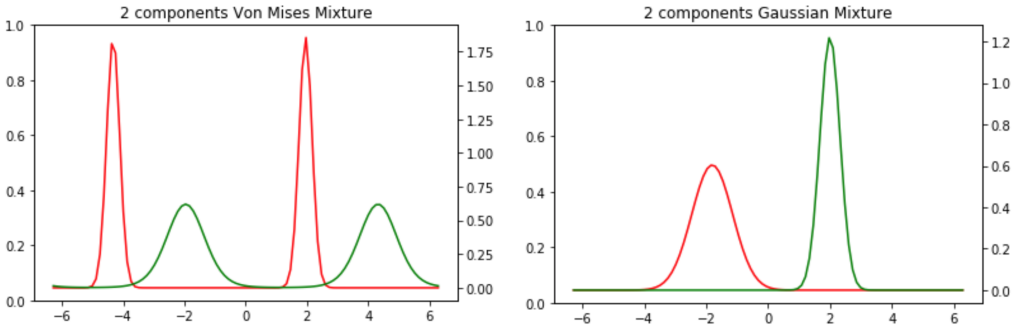
Figure 12 shows the results obtained by applying Von Mises Mixture on the transformed angular data. In 12a, different cluster of data is labeled with different color. And in 12b, it reveals that the distribution over θ has two components. Intuitively, the clusters from Von Mises Mixture model are more accurate than what we get from Gaussian Mixtures (Figure 9b).



(a) (b)

Figure 12: (a)Labeled plot of angular data θ with VMM. (b)Distribution over θ.

Moreover, if we extend the data interval, for example $\{-2\pi, 2\pi\}$, the Gaussian distribution does not hold true anymore while Von Mixes Mixture still gives us the correct result (Figure 13).



(a) (b)

Figure 13: Two components mixture models on angular data.

3.3 Optional additional questions

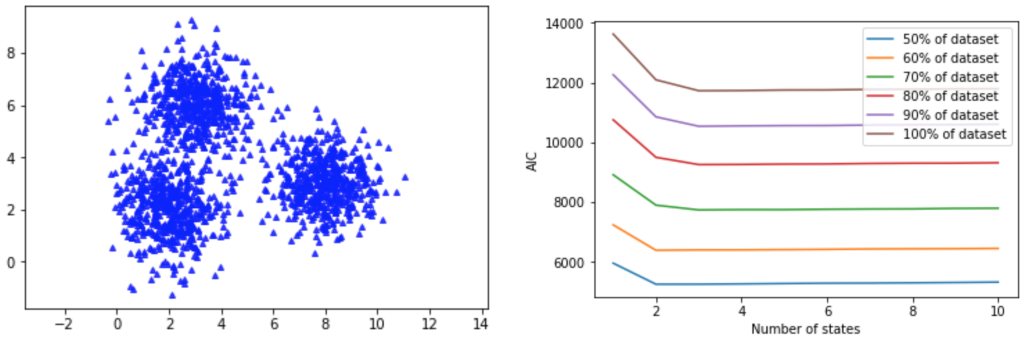
1. **Give a formal definition of consistent estimator of the number of states in a mixture model. Write some state-of-the-art on that topic, choose one of the references therein, justifying your choice. Provide a one-page description of the approach developed in that reference.**

   A consistent estimator of the number of states in a mixture model is the one that will provide the true optimal number of states for the mixture model when the training data size tends to infinity. This means that the distributions of the estimates become more and more concentrated near the true value of the parameter being estimated, so that the probability of the estimator being arbitrarily close to θ_0 converges to one.

2. **Imagine, describe and implement a protocol to evaluate consistency of any arbitrary estimator of the number of states. Test this protocol on mixtures of your choice, among mixtures of von Mises distributions and Gaussian mixtures.**

   Protocol: In short, for each size of training dataset, we investigate the AIC value when the number of states increases.

   We test this protocol on artificial data. We use mixture of Gaussian distributions in this case. As we can see in the Figure 14a, data forms three clusters. Now, we investigate the AIC values to find the optimal number of states.



(a) (b)

Figure 14: a) Raw data. b) AIC of GMM models with different number of states.

From figure 14b, we observe that when the training data is small, the optimal value for number of states is 2. However, when the size of training data increases, it yields consistently 3 for the number of states. It seems that our protocol works well. Figure 15 shows the final result when clustering the raw data.

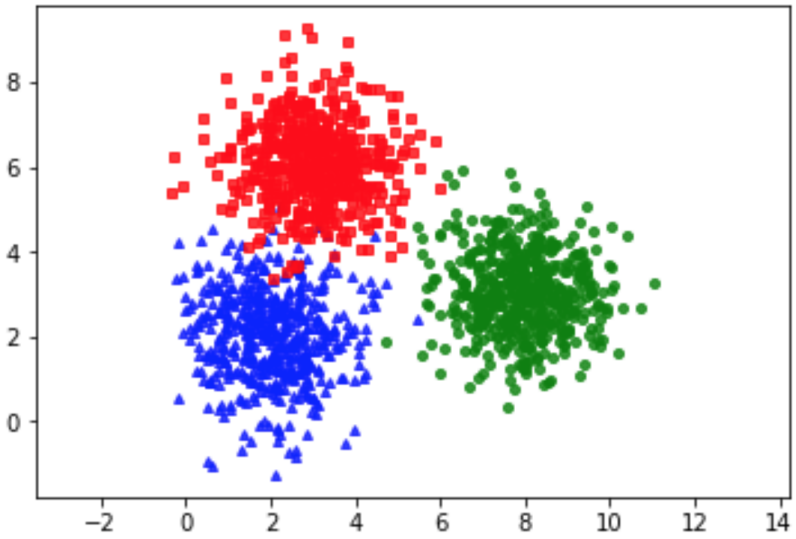


Figure 15: Labeled data with the optimal number of states.

3. **Apply the chosen approach to estimate the number of states in the real data set of part 3.1 (with mixture of von Mises distributions)? Does it yield the expected number of states?**

Figure 16 shows the varying AIC of mixture of von Mises distributions model. The optimal number of states corresponds to the one that minimizes the AIC value. When the size of training set increases, the optimal value for the number of states goes to 7. We conclude that we need 7 states for simulating letter $\mathcal{A}$.

4 Hidden Markov models

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

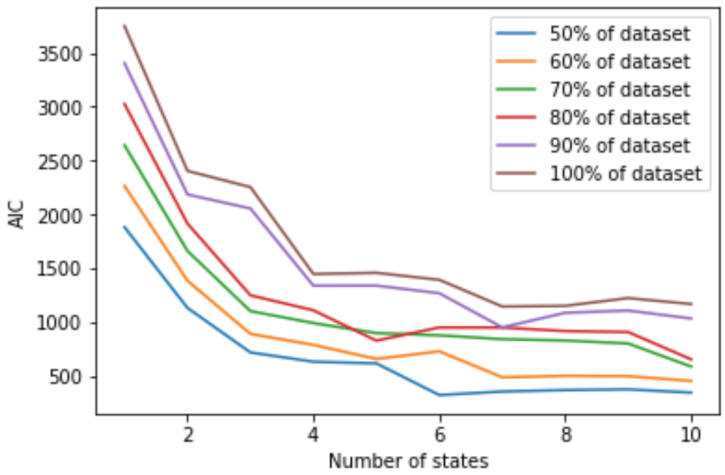


Figure 16: AIC value of mixture of von Mises distributions model when varying the size of training set.

4.1 Lab Work

Here you will focus on discriminating letters A and L, you may ignore the other four letters if you want.

1. **Define an HMC model (number of states, every parameter, ...) and simulate some trajectories, until it resembles letter Λ up to some Gaussian noise. Hint: you may use a differential representation of the signal. If you want to output $\mathbf{x_1}, ..., \mathbf{x_n}$, simulate $\mathbf{y_1} = \mathbf{x_1} - \mathbf{0}, \mathbf{y_2} = \mathbf{x_2} - \mathbf{x_1}, ..., \mathbf{y_2} = \mathbf{x_n} - \mathbf{x_{n-1}}$.**

   In this question we should think about what kind of HMC would generate a Λ. First, we suppose the letter is always drawn from left to right, so the number of states comes naturally as two: the first being the first half of the letter and the second being the second half. From our assumption, we also get that it should always start in the same state (the left half of the letter), so the starting probabilities are [1 0].

   Now we move to the transition matrix. To generate a letter Λ, we must have a small probability of going to the second state, but never go back. The values that we have chosen were:

$$\begin{bmatrix} 0.95 & 0.05 \\ 0 & 1 \end{bmatrix}$$

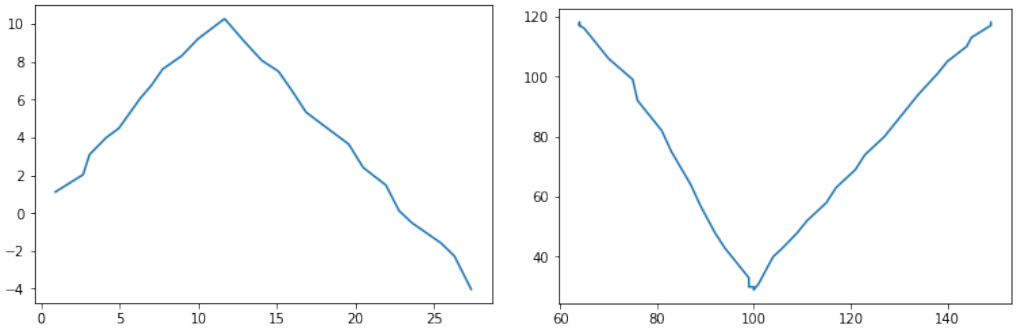
   As for the emission probabilities, the first state should point up and right, and the second should point down and right. It is also natural to then consider the means of the first state to be [1 1] and the second [1 -1]. For the covariance matrix, we tried a few different values to avoid too much noise and came up with the same one for both states:

$$\begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

2. **Develop some procedure to plot a real trajectory $\mathbf{X_{n_1}} = (\mathbf{x_1}, ..., \mathbf{xn}) \in (\mathbf{R^2})^{\mathbf{n}}$. Connect successive points with segments, but do not care about potential loss of temporal information in the produced figures. The trajectory will be a reverse letter but you may ignore this too. Include such figure in your report.**

   You can find the code in the attached Jupyter notebook. For a sample extracted from the model created in the last question we got Figure 17a and for file "a01" we got Figure 17b

3. **Normalize the trajectories by computing $\mathbf{y_t} = \frac{\mathbf{x_t} - \mathbf{x_{t-1}}}{\|\mathbf{xt} - \mathbf{xt-1}\|^2}$ for $\mathbf{t} = \mathbf{2}, ..., \mathbf{n}$. Estimate an HMC model with all the normalized trajectories for letter A, using bivariate**



(a) (b)

Figure 17

Gaussian emission distribution. Justify your choice for the number of states. Provide the estimates and comment them.

Again, the code comes attached in the Jupyter notebook. The number of states was chosen to be 2 for the same reason as before: it is the most intuive value, as we have explained. The estimates were:

Starting probabilities:

$$\begin{bmatrix} 0.12 & 0.88 \end{bmatrix}$$

Transition Matrix:

$$\begin{bmatrix} 0.9634 & 0.0366 \\ 0.1414 & 0.8586 \end{bmatrix}$$

Means:

$$\begin{bmatrix} -0.3738 & 0.8528 \\ -0.4449 & -0.8713 \end{bmatrix}$$

Covariances: state 1:

$$\begin{bmatrix} 0.0838 & 0 \\ 0 & 0.0493 \end{bmatrix}$$

state 2:

$$\begin{bmatrix} 0.0383 & 0 \\ 0 & 0.0045 \end{bmatrix}$$

We see that we the fitted model is very similar to what we thought before. Of course, the means have different signs because we trained the model with upside-down letters. Also, we thought about +-45 degrees slopes (since we used the same length for the x and y components), which is not what people actually do when writing the letter A. We can see that the mean values for the y-axis are bigger (in absolute value) than for the x-values, which means that the slopes of the line segments that compose the A's are more vertical than previously thought

4. **Same question as above for letter L. You may use the python hmmlearn or another library of your choice.**

Again, the number of states was chosen to be 2 and reasons are the same as before.

Starting probabilities:

$$\begin{bmatrix} 0.14 & 0.86 \end{bmatrix}$$

Transition Matrix:

$$\begin{bmatrix} 0.9484 & 0.0516 \\ 0.1413 & 0.8587 \end{bmatrix}$$

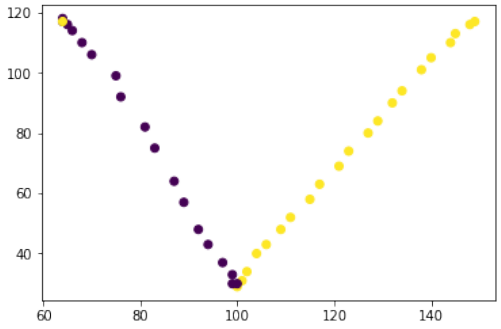


Figure 18

Means:

$$\begin{bmatrix} -0.8322 & 0.0687 \\ 0.009 & 0.9966 \end{bmatrix}$$

Covariances: state 1:

$$\begin{bmatrix} 0.223 & 0 \\ 0 & 0.0797 \end{bmatrix}$$

state 2:

$$\begin{bmatrix} 0.0067 & 0 \\ 0 & 0.0001 \end{bmatrix}$$

We can see again that the values for the means are what we naturally would expect: a practically vertical and a practically horizontal lines. The covariances in this case have also much smaller coefficients, which could be because the drawing of the letter L is simpler and less prone to noise. Also, for the letter A there is no "correct slope". People can draw A's in a lot of different ways and they would still be A's. The letter L is more constrained, given that it has to be composed of a vertical and a horizontal lines.

5. **Use the Viterbi algorithm on A1.txt and plot the unnormalized sequence using different colours for different states.**

   See Figure 18.

6. **Propose and implement a graphical (visual) method to validate the assumption of bivariate Gaussian emission distribution. What to think about this assumption?**

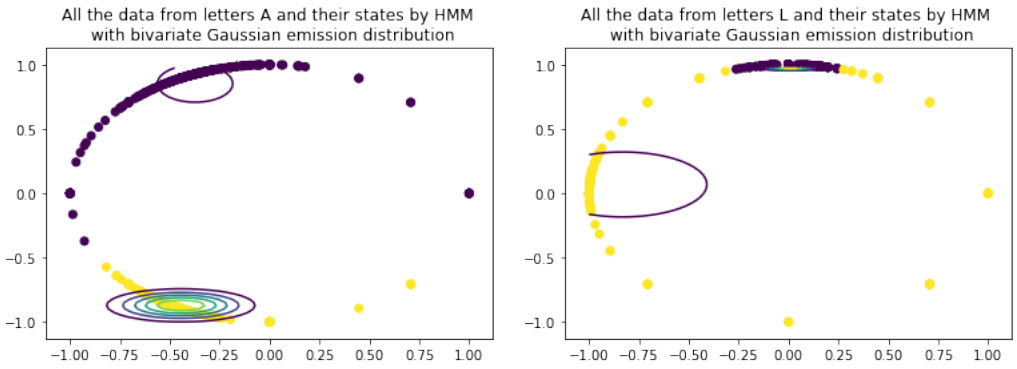
   A good way to evaluate the assumption is by looking at the scatter plot of the data with the colors indicating the states and also by looking at the X and Y histograms for the different states. On top of those, we overlapped the distributions given by the trained models and compared them.

   We can immediately see from Figure 19 that the distributions of the data do not look like Gaussian distributions. The closest is the purple class for the letter L. By looking at the histograms (Figures 20 and 21), in A we see patterns a bit closer to Gaussians , but for L they are extremely distinct. We therefore conclude that the assumption is not good.

4.2 Mandatory additional questions

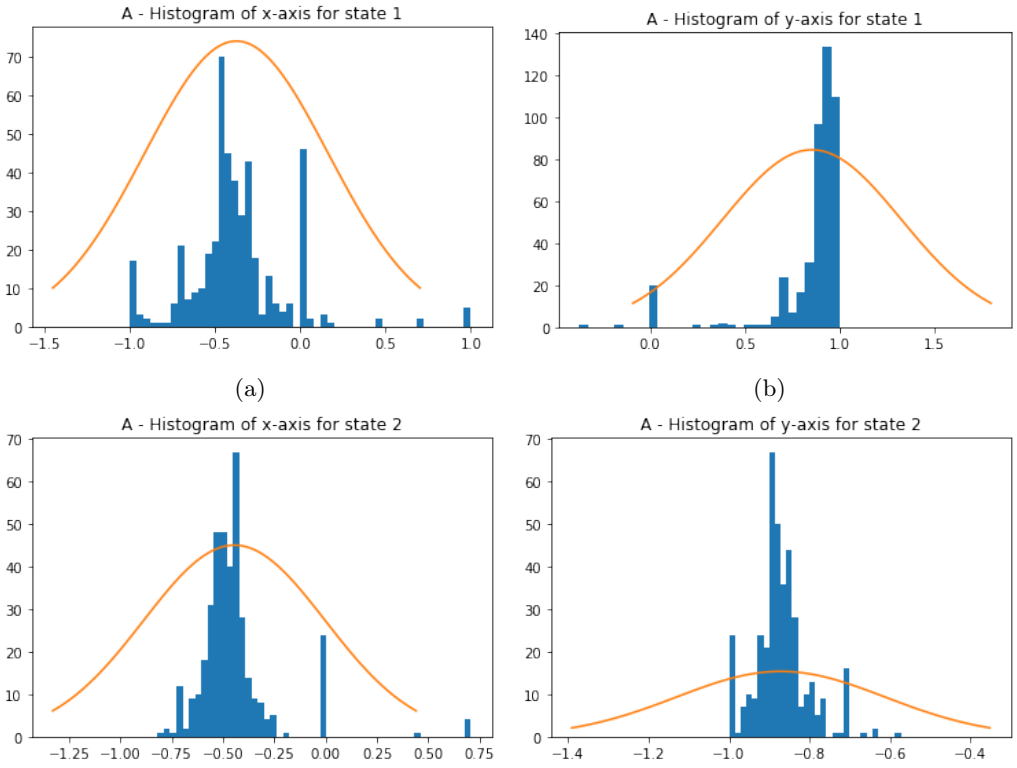
The aim of this part is to compare von Mises and Gaussian emission distributions.

1. **Transform the data to angular data. Implement von Mises emission distributions (including formal computations into the report) and compare the results with bivariate Gaussians.**



(a) (b)

Figure 19



(a) (b)

(c) (d)

Figure 20: Histogram for both states for letter A.

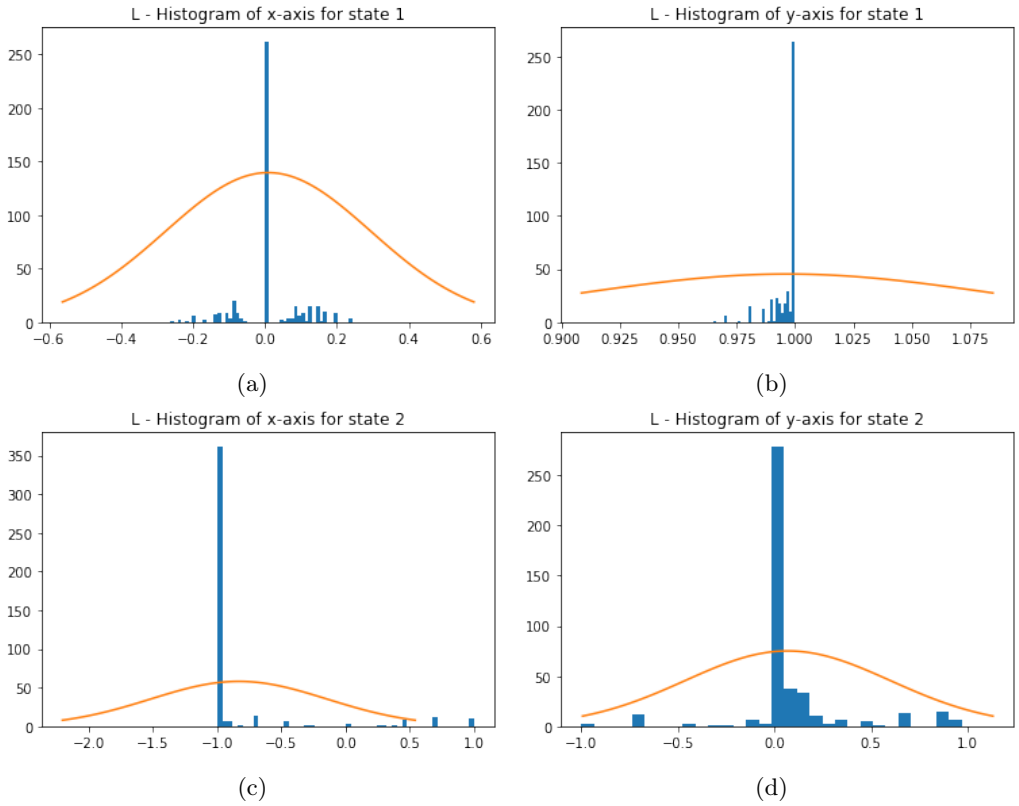


Figure 21: Histogram for both states for letter L.

Formal computations of von Mises HMM: From learning perspective, we could seek to find the parameters λ that maximize the completed likelihood $p(x_1^n, z_1^n)$. We use directly the explanation from lecture:

$$\begin{aligned}\hat{\lambda} = \arg\max_{\lambda} \mathbf{Q}(\lambda, \lambda^{(m)} &= \sum_{k=1}^{K} \ln(\pi_k) p_{\lambda^{(m)}}(z_1 = k|x_1^n) + \sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{t=2}^{n} \ln(a_{k,l}) p_{\lambda^{(m)}}(z_{t-1} = k, z_t = l|x_1^n) \\ &+ \sum_{k=1}^{K}\sum_{t=1}^{n} p_{\lambda^{(m)}}(z_t = k|x_1^n) \ln p_{\theta_k}(x_t) \\ &= \sum_{k=1}^{K} \ln(\pi_k)\gamma_1^{(m)}(k) + \sum_{k=1}^{K}\sum_{l=1}^{K}\sum_{t=2}^{n} \ln(a_{k,l})\gamma_t^{(m)}(k,l) + \sum_{k=1}^{K}\sum_{t=1}^{n} \ln \mathcal{V}(x_t; \mu_k, \kappa_k)\gamma_t^{(m)}(k) \\ s.t \qquad \forall k, \pi_k \geq 0, &\sum_{k=1}^{K} \pi_k = 1 \\ \forall l, a_{k,l} \geq 0, &\sum_{l} a_{k,l} = 1\end{aligned}$$

For **E-step**, compute $\gamma_t^{(m)}(j,k)$ and $\gamma_1^{(m)}(k)$ from $\beta_{t+1}(k)$ and $\alpha_t(j)$ as:

$$\begin{aligned}\gamma_t^{(m)}(j,k) = p(Z_t = j, Z_{t+1} = k|x_1^n) &= \frac{p(x_{t+1}^n|Z_{t+1} = k)}{p(x_{t+1}^n|x_1^t)} \alpha_{j,k} p(Z_t = j|x_1^t) \\ &= \beta_{t+1}(k) a_{j,k} \alpha_t(j) \\ \gamma_t^{(m)}(j) = p(Z_t = j|x_1^n) &= \sum_{k} p(Z_t = j, Z_{t+1} = k|x_1^n) = \sum_{k} \gamma_t^{(m)}(j,k)\end{aligned}$$

Where $\alpha_t(j)$ and $\beta_{t+1}(k)$ are computed from forward and backward procedures with von Mises emission distribution

$$\alpha_1(k) = \frac{p_{\theta_k}(x_1)\pi_k}{\sum_j p_{\theta_j}(x_1)\pi_j} = \frac{\pi_k \mathcal{V}(x_1; \mu_k, \kappa_k)}{\sum_j \pi_j \mathcal{V}(x_1; \mu_j, \kappa_j)}$$
$$\alpha_{t+1}(k) = \frac{\sum_j p_{\theta_k}(x_{t+1}) a_{j,k} \alpha_t(j)}{\sum_l \sum_k p_{\theta_k}(x_{t+1}) a_{l,k} \alpha_t(l)} = \frac{\sum_j \alpha_t(j) a_{j,k} \mathcal{V}(x_{t+1}; \mu_k, \kappa_k)}{\sum_l \sum_k \alpha_t(l) a_{l,k} \mathcal{V}(x_{t+1}; \mu_k, \kappa_k)}$$
$$\beta_n(k) = \frac{p_{\theta_k}(x_n)}{p(x_n|x_1^{n-1})} = \frac{\mathcal{V}(x_n; \mu_k, \kappa_k)}{p(x_n|x_1^{n-1})}$$
$$\beta_t(j) = \frac{\sum_k p_{\theta_j}(x_t) a_{j,k} \beta_{t+1}(k)}{p(x_t|x_1^{t-1})} = \frac{\sum_k a_{j,k} \beta_{t+1}(k) \mathcal{V}(x_t; \mu_j, \kappa_j)}{p(x_t|x_1^{t-1})}$$

For **M-step**, we construct the Langarian from objective function (maximizing completed log-likelihood):

$$\mathcal{L}(\lambda, \eta, \nu) = \sum_{k=1}^{K} \ln(\pi_k) \gamma_1^{(m)}(k) + \sum_{k=1}^{K} \sum_{l=1}^{K} \sum_{t=2}^{n} \ln(a_{k,l}) \gamma_t^{(m)}(k,l) + \sum_{k=1}^{K} \sum_{t=1}^{n} \gamma_t^{(m)}(k) \ln \mathcal{V}(x_t; \mu_k, \kappa_k)$$
$$+ \eta(1 - \sum_{k=1}^{K} \pi_k) + \sum_{k=1}^{K} \nu_k (1 - \sum_{l=1}^{K} a_{k,l})$$

Taking partial derivatives and setting them equal to zero we get:

$$\pi_k = \frac{\gamma_1^{(m)}(k)}{\sum_{j=1}^{K} \gamma_1^{(m)}(j)}$$
$$a_{k,l} = \frac{\sum_{t=2}^{n} \gamma_t^{(m)}(k,l)}{\sum_{l=1}^{K} \sum_{t=2}^{n} \gamma_t^{(m)}(k,l)}$$
$$\mu_k = \arctan\left(\frac{\sum_{t=1}^{n} \gamma_t^{(m)}(k) \sin x_t}{\sum_{t=1}^{n} \gamma_t^{(m)}(k) \cos x_t}\right)$$
$$A(\kappa_k) = \frac{\sum_{t=1}^{n} \gamma_t^{(m)}(k) \cos(x_t - \mu_k)}{\sum_{t=1}^{N} \gamma_t^{(m)}(k)}$$

The value of κ_k can be computed by approximating the value of $A_2^{-1}(x)$ by

$$A_2^{-1}(x) \approx \frac{2x - x^3}{1 - x^2}$$

Now, moving to the numerical results:

Results for letter A

Starting probabilities:

$$\begin{bmatrix} 0.1946 & 0.8054 \end{bmatrix}$$

Transition Matrix:

$$\begin{bmatrix} 0.86 & 0.14 \\ 0.0338 & 0.9662 \end{bmatrix}$$

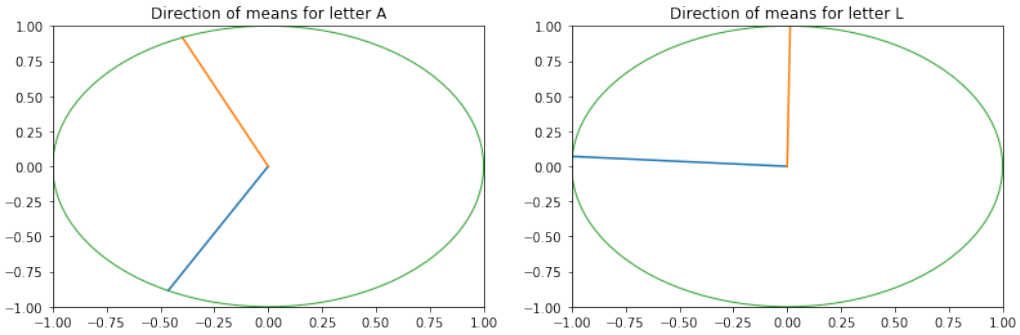
Means:

$$\begin{bmatrix} -2.0535 & 1.9802 \end{bmatrix}$$

Kappas:

$$\begin{bmatrix} 34.55 & 6.13 \end{bmatrix}$$

Results for letter L



(a) (b)

Figure 22

Starting probabilities:

$$\begin{bmatrix} 0.8197 & 0.1803 \end{bmatrix}$$

Transition Matrix:

$$\begin{bmatrix} 0.9736 & 0.0264 \\ 0.1202 & 0.8798 \end{bmatrix}$$

Means:

$$\begin{bmatrix} 3.0709 & 1.5568 \end{bmatrix}$$

Kappas:

$$\begin{bmatrix} 3.65 & 106.20 \end{bmatrix}$$

Figure 22 gives us a more visual understanding of the mean angles found. It is important to note that they are slightly distorted because the x and y axis have different sizes (we see ellipses instead of circles).

2. **Compute a 5-fold cross-validated classification error with both families of emission distributions (for the moment, only consider classes A and L.)**

   For both distributions the error was 0 in all folds. Please see the code in the attached notebook.

4.3 Optional additional questions

1. **Give a formal definition of consistent estimators of the number of states in a HMM. 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**

   A consistent estimator of the number of states in a HMM one that, under infinite data, will provide the true optimal number of parameters for the HMM. We were not able or did not have time to go further and pick a consistent method. Instead, we decided to use a popular method for the following questions: BIC.

2. **Imagine, describe and implement a protocol to evaluate consistency of any arbitrary estimator of the number of states. Test this protocol on HMM with von Mises distributions, using the estimator chosen in the previous question**

   We unfortunately did not have time to implement the method, so we decided to provide an high-level idea of what we would have done.

   We must find out if our estimator will find the true number of states under infinite data. To do so, we can sample data of increasing size from an artificial HMM and apply the method to it. In a more structured way:

   (a) Pick a number of states s

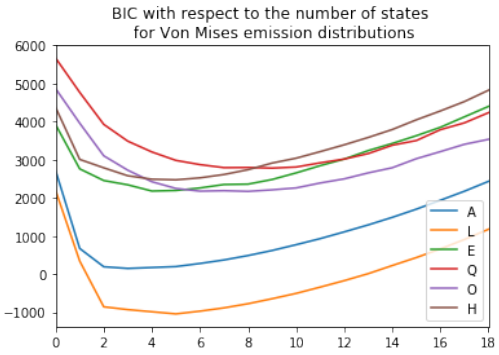


Figure 23

(b) Create an artificial HMM with Von Mises distributions with s states and random parameters

(c) Generate samples of increasing size

(d) Apply the method in all the samples

(e) Observe the asymptotic behavior of the estimator

(f) Repeat increasing the number of states

3. **Apply the chosen approach to estimate the number of states in the real data set of part 4.1 (with mixture of von Mises distributions), considering now the 6 letters A, E, H, L, O and Q.**

   We we mentioned before, we have chosen BIC as the metric to choose the number of parameters. The number of parameters found can be seen in Table 1. We also added a plot (Figure 23) showing the relation between BIC and the number of states for the 6 letters.

Letter	Number of States
A	4
L	6
E	5
Q	10
O	9
H	6

Table 1: Optimal number of states returned by BIC

4. **Compute a 5-fold cross-validated classification error using the 6 estimated models. Provide the parameter estimates for each of the 6 models. Provide some confusion matrix using the selected models and comment the errors.**

   The cross-validation accuracy was of around 94%. The confusion matrix can be seen in Figure 24.

   As we can see, most of the errors were made by mispredicting E as Q. This was unexpected because the most visually similar letters are Q and O. We took a look at the mean angles for E and Q to look for insights. As we can see in Figure 25, all the angles in E have very close corresponding angles in Q. This could be partially explained by the fact that Q has many states, so it is probable the they will be close to some of E, but we believe that such explanation is not sufficient.

   It is interesting to note that, if we change the number of states in Q to 5, this problem is practically solved and we get an cross-validation accuracy of 98%. If we, again, look at the mean angles, as in Figure 26, we see that the they are less similar now.

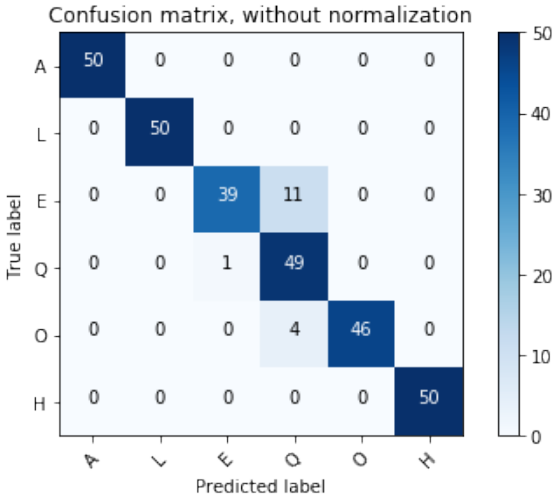
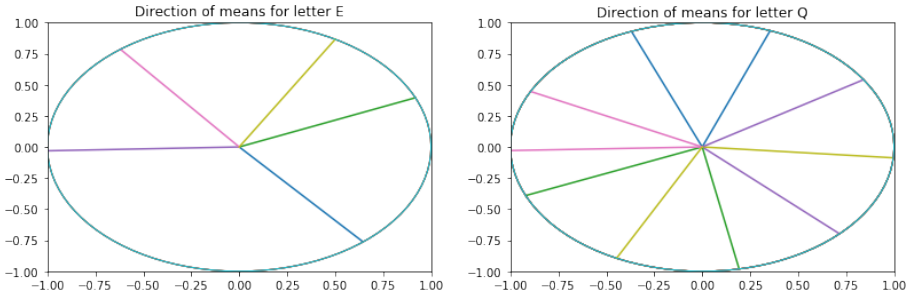
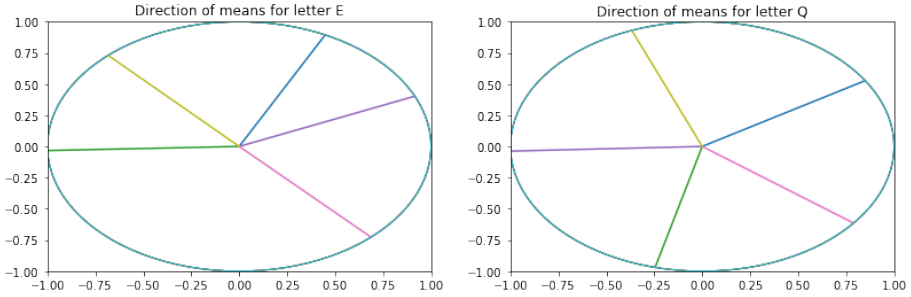


Figure 24



(a) (b)

Figure 25: Mean angles for Q and E with 10 and 5 states, respectively.



(a) (b)

Figure 26: Mean angles for Q and E with 5 states.

5 Conclusion

Throughout the whole project, we have a deeper understanding of this course, both theoretically and practically. This project was an opportunity for us to put into practice our knowledge that we have learned in classes. The freedom of choice we have enjoyed and the autonomy required for the realization of this project are elements that allowed us to evolve in a coordinated way while maintaining a good group dynamic. Despite the difficulties encountered, this project was an opportunity to grasp the dimensions of research topics; from the design to the implementation of solutions, through the organization and communication within the team.

References

[1] C. Glymour P. Spirtes and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, 2000.

[2] M Kalisch and P Buhlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(8):613–636, 2007.