Statistical Inference and Machine Learning

MGT-448

Homework 3

PCA and EM Algorithm

Students
Student1 Alexis Couturier
Student2 Erick Maraz Zuniga
Student3 Alon Tchelet
Student4 Mustafa Yildirim

Professor
Negar Kiyavash

Abstract

Third homework assignment of MGT-448: Statistical Inference and Machine Learning course covering the topics of PCA and EM Algorithms.

January 31, 2022

1 PCA

1. The objective of *Principal Component Analysis* (PCA) is to find a lower dimension subspace which still preserves the richness of the data i.e find a set of vectors to project the data on such that variance is maximized.

   The first step in PCA is to standardize dataset $D = \{x_1, x_2, ..., x_n\}$ where $x_i \in \mathbb{R}^d$, n is the number of data samples and d the number of features. For a given observation $x_i \in \mathbb{R}^d$, the mean feature vector μ (Eq.1) and standard deviation feature vector σ (Eq.2) are used to standardize the data point :

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{2}$$

$$x_i' = \frac{x_i - \mu}{\sigma} \tag{3}$$

   Applying this process to each data point in D returns $D' = \{x_1', x_2', ..., x_n'\}$, a set of standardized observations.

   Finding a direction u on which to project data D' such that variance is maximized translates into the following optimization problem :

$$\max_{u \in \mathbb{R}^d} u^T \Sigma u \quad \text{s.t} \quad ||u||_2 = 1 \tag{4}$$

   where Σ is the covariance matrix associated with set D'. From the augmented Lagrangian problem, one finds that $u \in \mathbb{R}^d$ has to solve for $\Sigma u = \lambda u$ i.e it has to be the eigenvector of Σ associated with the largest eigenvalue.

   The next step is to calculate covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$ for the standardized observations. Lets denote by $X' \in \mathbb{R}^{n \times d}$ the matrix of observations associated with dataset D'. The covariance matrix Σ is defined as:

$$\Sigma = cov[X', X'] = \mathrm{E}[(X' - \mu_{X'})^T (X' - \mu_{X'})] \tag{5}$$

   where $E[\cdot]$ is the expected value function and $\mu_{X'} = h^T \mu \in \mathbb{R}^{n \times d}$ the mean feature matrix of X' with $h = (1, 1, ...1) \in \mathbb{R}^{1 \times n}$.

   The eigenvalues of Σ are computed using Python's *np.linalg.eig()* command which returns both eigenvalues and their associated eigenvectors. The graph below shows the eigenvalues ranked in ascending order :

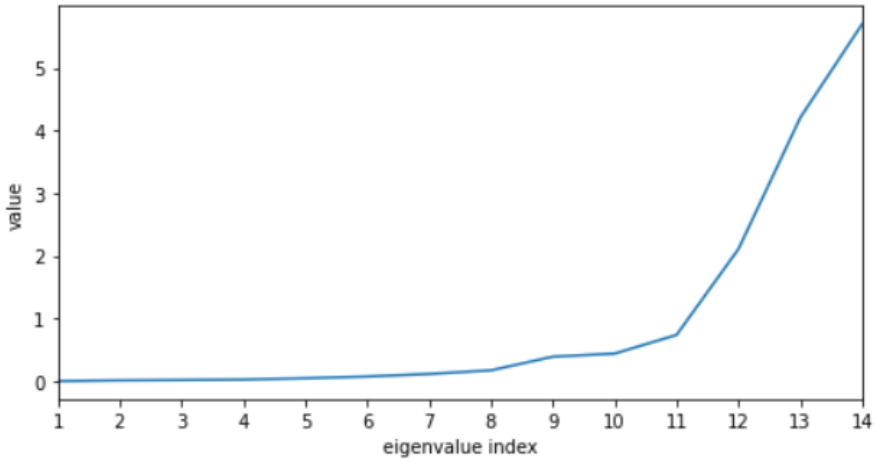


Figure 1: A plot of the eigenvalues in ascending order

2. From Figure 1, one can clearly see that the first 8 eigenvalues are very small and hence have a very low influence of the data covariance. By normalizing the eigenvalues as in Eq. 6, one can assess the percentage of variance dictated by each eigenvalue.

$$\lambda' = \frac{\lambda}{\sum_{i=1}^{d} \lambda_i} \tag{6}$$

where $\lambda' \in \mathrm{R}^d$ is the normalized eigenvalue vector representing the percentage of covariance covered by each eigenvalue in $\lambda \in \mathbb{R}^d$.
Performing a cumulative sum on the first k values of λ' in descending order, one can choose a value of k such that the covered variance is above a given threshold.

k	1	2	3	4	5	6	7
covered %	40.6%	70.6%	85.6%	90.8%	93.9%	96.7%	97.9%
k	8	9	10	11	12	13	14
covered %	98.7%	99.3%	99.6%	99.8%	99.91%	99.99%	100%

Table 1: Percentage of variation covered by different values of k

After the examination of Table 1, $k = 6$ was chosen as it reduces the features space by more than half while maintaining more than 95% of the data variation.

3. The eigenvectors for $k = 2$ are:

$$w_1 = \begin{bmatrix} -0.093812 \\ -0.192439 \\ 0.538282 \\ 0.129275 \\ -0.172358 \\ 0.617888 \\ -0.009354 \\ -0.003232 \\ -0.225518 \\ -0.413895 \\ 0.103254 \\ 0.055689 \\ -0.013636 \\ -0.000227 \end{bmatrix}, w_2 = \begin{bmatrix} -0.190165 \\ -0.025256 \\ 0.519236 \\ -0.168502 \\ 0.463153 \\ -0.470015 \\ -0.177409 \\ -0.371746 \\ -0.179096 \\ -0.128162 \\ -0.011042 \\ -0.102907 \\ -0.029918 \\ -0.000145 \end{bmatrix} \tag{7}$$

4. In order to plot Figure 2, α_1 and α_2 needed to be computed. As explained in the outline, α_j is computed as the dot product of the original feature vectors $x_i \in \mathrm{R}^d$ and the respective eigenvector $w_j \in \mathrm{R}^d$ as shown in Eq. 8.

$$\alpha_j = w_j^T x_i \tag{8}$$

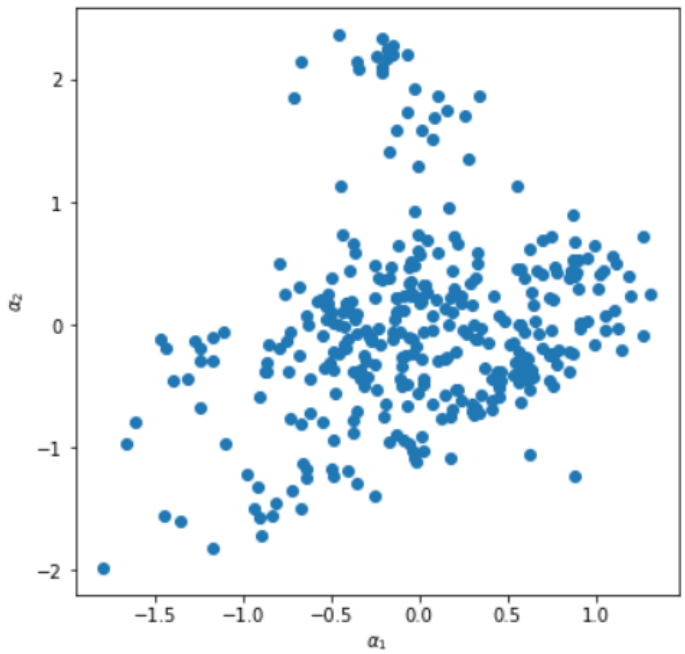


Figure 2: Scatter plot of the data in terms of α_1 and α_2

5. The PCA estimated data $\tilde{x}_i$ is estimated as follows :

$$\tilde{x}_i = \sum_{j=1}^{k} w_j^T x_i' w_j \tag{9}$$

where w_1 and w_2 are given in question 3. This step is performed on every data point in D'.

The mean squared recombination error is computed as follows :

$$MSRE = \frac{1}{n} \sum_{i=1}^{n} ||x_i' - \tilde{x}_i||_2^2 \tag{10}$$

Using the above expression, the MSRE was found to be 0.9286.

2 EM Algorithm

2.1 Task 1 - EM for Mixture of Multinomials

1. Let's verify the above-derived E-step and M-step.

   E-step :

   In this step the algorithm computes $Q(c) = P(c|D_i; \mu, \pi)$. According to Bayes' theorem and using the probability distributions from the outline, this leads to :

$$Q(c) = P(c|D_i; \mu, \pi) = \frac{P(D_i|c)P(c)}{P(D_i)} = \frac{\prod_{j=1}^{n_w} \mu_{jc}^{T_{i,j}} \times \pi_c}{\sum_{c=1}^{n_c} \pi_c \prod_{j=1}^{n_w} \mu_{jc}^{T_{i,j}}} \tag{11}$$

   E-Step :

   In this step the algorithm computes :

$$\underset{\mu,\pi}{\text{argmax}} \sum_{i=1}^{n_d} \text{ELBO}(D_i; Q(c), \mu, \pi) = \underset{\mu,\pi}{\text{argmax}} \sum_{i=1}^{n_d} \sum_{c=1}^{n_c} Q(c) \log \frac{P(D_i, c; \mu, \pi)}{Q(c)} \tag{12}$$

   Using the chain rule $P(D_i, c; \mu, \pi) = P(D_i|c; \mu, \pi)P(c)$ one can write the objective function as :

$$\mathcal{L}(\mu, \pi) = \sum_{i=1}^{n_d} \sum_{c=1}^{n_c} \gamma_{ic} \log \frac{\prod_{j=1}^{n_w} \mu_{jc}^{T_{i,j}} \cdot \pi_c}{\gamma_{ic}} \tag{13}$$

   Two constraints on the probability distributions of $\mu_{|c}$ and π can be written :

$$\text{Constraint 1 : } \sum_{c=1}^{n_c} \pi_c = 1 \quad \text{and} \quad \text{Constraint 2 : } \sum_{j=1}^{n_w} \mu_{jc} = 1 \tag{14}$$

   Using the first constraint in a Lagrange multiplier, one can write :

$$\bar{\mathcal{L}}(\mu, \pi) = \mathcal{L}(\mu, \pi) + \lambda \left(1 - \sum_{c=1}^{n_c} \pi_c\right) \tag{15}$$

   Considering γ_{ic} as a constant set in the E-step (i.e independent of π_c), taking the derivative with respect to π_m and equating to zero, one gets :

$$\frac{\partial \bar{\mathcal{L}}}{\partial \pi_m} = 0 \iff \sum_{i=1}^{n_d} \gamma_{im} \cdot \frac{1}{\pi_m} - \lambda = 0 \iff \pi_m = \frac{1}{\lambda} \sum_{i=1}^{n_d} \gamma_{im} \tag{16}$$

   Using the first constraint and summing the above equality over m :

$$\sum_{m=1}^{n_c} \frac{1}{\lambda} \sum_{i=1}^{n_d} \gamma_{im} = 1 \iff \sum_{i=1}^{n_d} \frac{1}{\lambda} \sum_{m=1}^{n_c} \frac{p\,(D_i, m)}{p(D_i)} = 1 \iff \frac{n_d}{\lambda} = 1 \iff \lambda = n_d \tag{17}$$

Finally, looking back at equation (16) the same result as in the outline is obtained :

$$\pi_c = \frac{1}{n_d} \sum_{i=1}^{n_d} \gamma_{ic} \tag{18}$$

Using the second constraint in a Lagrange multiplier, one can write :

$$\bar{\mathcal{L}}(\mu, \pi) = \mathcal{L}(\mu, \pi) + \lambda \left(1 - \sum_{j=1}^{n_w} \mu_{ic} \right) \tag{19}$$

Taking the derivative with respect to μ_{lp} and equating to zero, one gets :

$$\frac{\partial \bar{\mathcal{L}}}{\partial \mu_{lp}} = 0 \iff \sum_{i=1}^{n_d} \gamma_{ip} \cdot T_{il} \cdot \frac{1}{\mu_{lp}} - \lambda = 0 \iff \mu_{lp} = \frac{1}{\lambda} \sum_{i=1}^{n_d} \gamma_{ip} T_{il} \tag{20}$$

Using the first constraint and summing the above equality over l :

$$\sum_{l=1}^{n_\omega} \frac{1}{\lambda} \sum_{i=1}^{n_d} \gamma_{ip} T_{il} = 1 \iff \lambda = \sum_{l=1}^{n_\omega} \sum_{i=1}^{n_d} \gamma_{ip} T_{il} \tag{21}$$

Finally, looking back at equation (20) the same result as in the outline is obtained :

$$\mu_{jc} = \frac{\sum_{i=1}^{n_d} \gamma_{ic} T_{ij}}{\sum_{j=1}^{n_w} \sum_{i=1}^{n_d} \gamma_{ic} T_{ij}} \tag{22}$$

2. The algorithm is implemented using Python and uses the following initialization parameters :

$$\pi^{(0)} = (1/n_c, 1/n_c, ..., 1/n_c) \tag{23}$$

$$\mu^{(0)} \in \mathbb{R}^{n_w \times n_c} \text{ has its coefficients drawn from a uniform distribution over [0,1]} \tag{24}$$

The EM-algorithm then runs for 50 iterations. Due to the randomness in the initialization of μ the accuracy varies depending on the initial choice of $\mu^{(0)}$. In order to have a better understanding on how the algorithm performs on average, the program was run a 100 times and the accuracy scores saved in a list. Plotting an histogram of the accuracy, one gets :

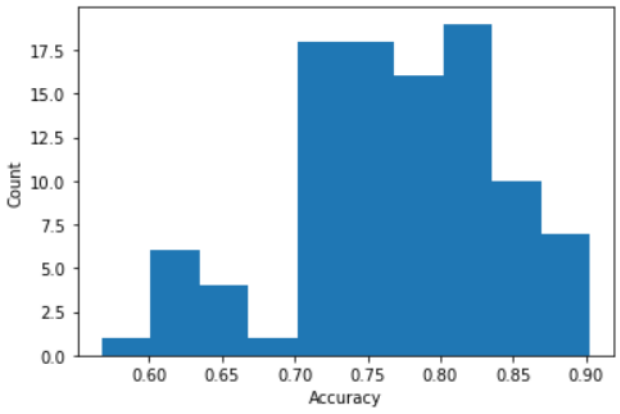


Figure 3: 100 values of accuracy and their count

The mean accuracy is 76.8% and 88% of the values lie above 0.7. Using the empirical distribution of μ to initialize, the program returns an accuracy of 92.75%.

Looking at the empirical distribution of π, it turns out to be identical to the naive initialising vector chosen above. As it seems to be a reasonable and standard choice, no further initialising methods were tested.

```
predicted label 0 is most likely label 1 [100   6   0   0]
predicted label 1 is most likely label 2 [ 0 71  0  0]
predicted label 2 is most likely label 3 [  0  14 100   0]
predicted label 3 is most likely label 4 [  0   9   0 100]
The accuracy of the EM Algorithm is 0.9275
```

Figure 4: Example of output from the code

2.2 Task 2 - MCMC Algorithm

1. • Let's assume that $\rho \sim Dirichlet(\delta_1, ..., \delta_K)$ and estimate posterior $\rho|x, z$.

   Given :

$$\mathcal{L} = \prod_{k=1}^{K} \rho_k^{n_k} \left(\prod_{i:z_i=k} \mathcal{N}_k \left(x_i \mid \mu_k, \phi_k \right) \right), \quad \sum_{k=1}^{K} n_k = n \quad \text{and} \quad n_k = Card\left\{ z_i = k \right\} \tag{25}$$

   Using Bayes' rule one gets :

$$p(\rho \mid x, z) \propto p(x, z|\rho) \cdot p(\rho) \propto \mathcal{L} \cdot p(\rho) = \prod_{k=1}^{K} \rho_k^{n_k} \left(\prod_{i:z_i=k} \mathcal{N}_k \left(x_i \mid \mu_k, \phi_k \right) \right) \cdot p(\rho) \tag{26}$$

   Variable ρ has a Dirichlet distribution :

$$p(\rho \mid x, z) \propto \prod_{k=1}^{K} \rho_k^{n_k} \left(\prod_{i:z_i=k} \mathcal{N}_k \left(x_i \mid \mu_k, \phi_k \right) \right) \cdot \prod_{k=1}^{K} \rho_k^{\delta_k - 1} \propto \prod_{k=1}^{K} \rho_k^{n_k + \delta_k - 1} \tag{27}$$

   This shows that posterior $\rho|x, z \sim Dirichlet(\delta_1^*, ..., \delta_K^*)$ where $\forall k \in [\![1; K]\!],\ \delta_k^* = n_k + \delta_k$.

   • Let's assume that $\phi_k \sim Gamma(a/2, b/2)$ and estimate posterior $\phi_k|x, z$. Using Bayes' rule one gets :

$$p(\phi \mid x, z) \propto p(x, z|\phi) \cdot p(\phi) \propto \mathcal{L} \cdot p(\phi) = \prod_{k=1}^{K} \rho_k^{n_k} \left(\prod_{i:z_i=k} \mathcal{N}_k \left(x_i \mid \mu_k, \phi_k \right) \right) \cdot p(\phi_k) \tag{28}$$

   Re-writing the normal distribution in exponential form :

$$\mathcal{L} \propto \prod_{k=1}^{K} \rho_k^{n_k} \cdot \phi_k^{\frac{n_k}{2}} \exp\left(-\frac{\phi_k}{2} \sum_{i:z_i=k} \left(x_i - \mu_k \right)^2 \right) \tag{29}$$

   Using the fact that ϕ_k has a gamma distribution :

$$p\left(\phi \mid x, z\right) \propto \prod_{k=1}^{K} \rho_k^{n_k} \cdot \phi_k^{n_k/2} \exp\left(-\frac{\phi_k}{2} \sum_{i:z_i=k} \left(x_i - \mu_k \right)^2 \right) \cdot \phi_k^{\frac{a}{2}-1} \cdot e^{-\frac{b\phi_k}{2}} \tag{30}$$

   Which can be rewritten as :

$$p\left(\phi \mid x, z\right) \propto \prod_{k=1}^{K} \phi_k^{(n_k+a)/2-1} \exp\left(-\frac{\phi_k}{2} \left(b + \sum_{i:z_i=k} \left(x_i - \mu_k \right)^2 \right) \right) \tag{31}$$

   This shows that posterior $\phi_k|x, z \sim Gamma(a_k^*/2, b_k^*/2)$ where $\forall k \in [\![1; K]\!],\ a_k^* = n_k + a$ and $b_k^* = b + \sum_{i:z_i=k} \left(x_i - \mu_k \right)^2$.

   • Let's assume that $\mu_k|\phi_k \sim \mathcal{N}(m_k, 1/\alpha_k\phi_k)$ and estimate posterior $\mu_k|x, z, \phi_k$. Using Bayes' rule one gets :

$$p\left(\mu \mid x, z, \phi\right) \propto p\left(x, z, \phi| \mu\right) \cdot p\left(\mu \mid \phi\right) = \mathcal{L} \cdot p\left(\mu \mid \phi\right) \tag{32}$$

   Re-writing the normal distribution in exponential form and using the fact that $\mu_k|\phi_k$ is normally distributed :

$$\mathcal{L} \cdot p\left(\mu \mid \phi\right) \propto \prod_{k=1}^{K} \exp\left(-\frac{\phi_k}{2} \sum_{i:z_i=k} \left(x_i - \mu_k\right)^2\right) \cdot \exp\left(\frac{-\alpha_k \phi_k}{2} \left(\mu_k - m_k\right)^2\right) \tag{33}$$

Using $\exp(a)\exp(b) = \exp(a+b)$ and expanding both $(x_i - \mu_k)^2$ and $(\mu_k - m_k)^2$ in the first and second term of the summation :

$$= \prod_{k=1}^{K} \exp\left(\mu_k^2 \underbrace{\left(-\frac{\alpha_k \phi_k}{2} - \frac{\phi_k}{2} \sum_{i:z_i=k} 1\right)}_{(1)} - 2\mu_k \underbrace{\left(-\frac{\alpha_k \phi_k}{2} m_k - \frac{\phi_k}{2} \sum_{i:z_i=k} x_i\right)}_{(2)} - \frac{\phi_k}{2} \underbrace{\left(m_k^2 + \sum_{i:z_i=k} x_i^2\right)}_{(3)}\right) \tag{34}$$

By definition $\mu_k | x, z, \phi_k \sim \mathcal{N}(m_k^*, \frac{1}{\phi_k \alpha_k^*})$ and therefore :

$$p(\mu | x, z, \phi) = \prod_{k=1}^{K} \exp\left(-\frac{\alpha_k^* \phi_k}{2} \left(\mu_k - m_k^*\right)^2\right) = \prod_{k=1}^{K} \exp\left(-\frac{\alpha_k^* \phi_k}{2} \left(\mu_k^2 - 2\mu_k {m_k}^* + {m_k^*}^2\right)\right) \tag{35}$$

By identification :

$$\text{On term (1) :} \quad -\frac{\alpha_k^* \phi_k}{2} = -\frac{\alpha_k \phi_k}{2} - \frac{\phi_k}{2} \sum_{i:z_i=k} 1 \implies \alpha_k^* = \alpha_k + n_k \tag{36}$$

$$\text{On term (2) :} \quad -\frac{\alpha_k^* \phi_k}{2} \cdot m_k^* = -\frac{\alpha_k \phi_k}{2} m_k - \frac{\phi_k}{2} \sum_{i:z_i=k} x_i \implies m_k^* = \frac{1}{\alpha_k^*} \left(m_k \alpha_k + \sum_{i:z_k=k} x_i\right) \tag{37}$$

Using the above expression on α_k^* one gets :

$$m_k^* = \frac{m_k \alpha_k + n_k \bar{x}_k}{\alpha_k + n_k} \text{ where } \bar{x}_k = \frac{1}{n_k} \sum_{i:z_i=k} x_i \tag{38}$$

This shows that posterior $\mu_k | x, z, \phi_k \sim \mathcal{N}(m_k^*, \frac{1}{\alpha_k^* \phi_k})$ where $\forall k \in [\![1; K]\!]$, $\alpha_k^* = n_k + \alpha_k$ and $m_k^* = \frac{m_k \alpha_k + n_k \bar{x}_k}{\alpha_k + n_k}$ with $\bar{x}_k = \frac{1}{n_k} \sum_{i:z_i=k} x_i$.

2. While implementing the MCMC algorithm, some considerations were made to get good initial values for the unknown distribution parameters. These parameters were randomly chosen again by the following criteria.

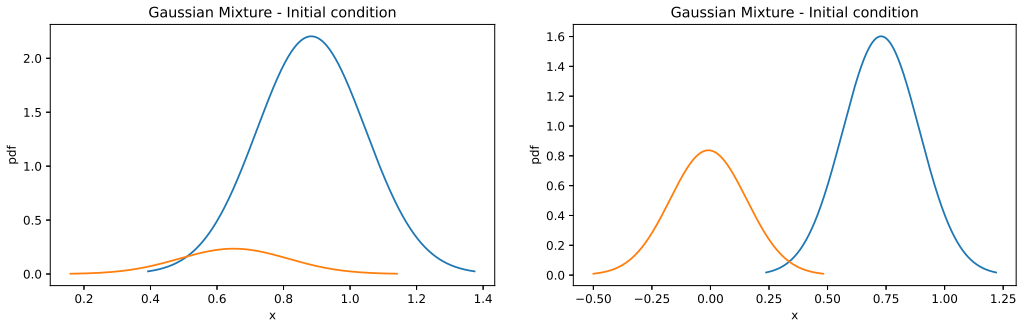
   (a) The means of the distributions μ_k should be in the range of the data.

   (b) The inverse of the variance ϕ_k should be close to the precision of the data.

   (c) The sum of the the weights ρ_k should sum up to one (i.e. $\sum_{k=0}^{K} \rho_k = 1$). This condition is always satisfied by the Dirichlet distribution.

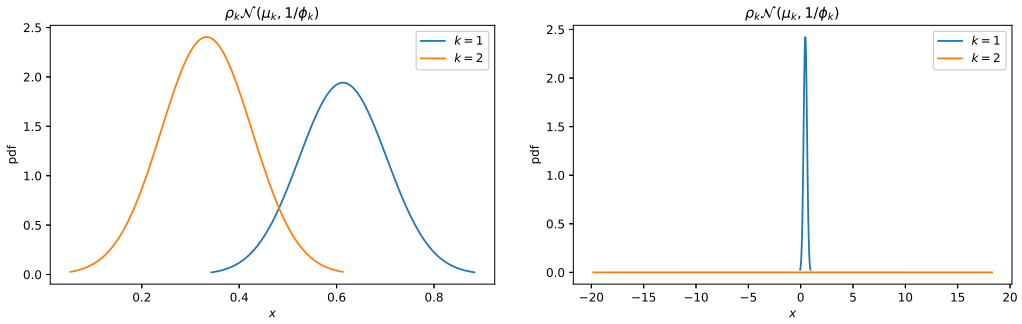
   (d) Iterate for new random values until $n_k \geq \frac{1}{2K} \quad \forall k = 1, \ldots, K$. This condition is an heuristic found while testing the algorithm. The intuition is that the initial values of the distribution should already divide the points in different clusters.

   However, despite these initial conditions, the algorithm does not always converge (most of the time it converges). It was observed that this often happens when one initial mean μ_k is far way from the data mean and the other is very close to the data mean. Additional heuristics and(or) conditions can be made to ensure that the algorithm will always converge. Figure 5 shows initial conditions that led to a successful (left) and unsuccessful (right) clustering and Figure 6 shows the distributions with the parameters computed.



(a) (b)

Figure 5: Pdf with initial parameters which led to (a) successful and (b) unsuccessful clustering.



(a) (b)

Figure 6: Pdf with the computed parameters, after running the algorithm. (a) Successful and (b) unsuccessful clustering.

The posterior distribution of the unknown parameters for the successful case is shown in Figure 7 and their posterior mean, in Table 2.

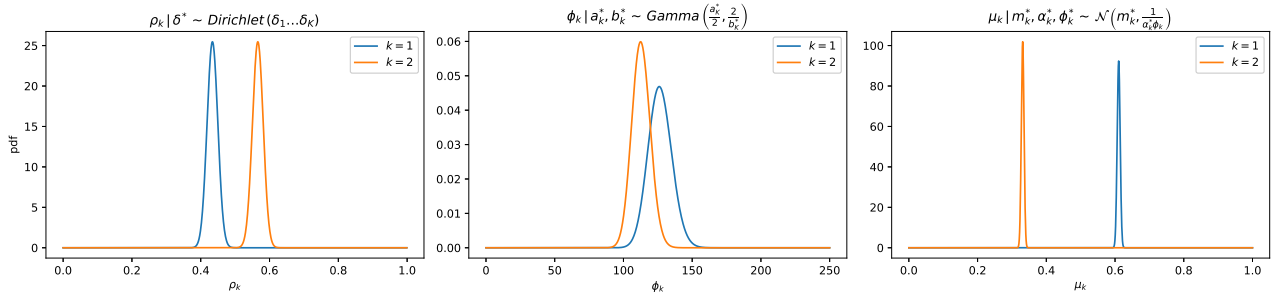


Figure 7: Pdf's of the computed parameters

Parameter	ρ_1	ρ_2	ϕ_1	ϕ_2	μ_1	μ_2
mean-1000 samples	0.43443	0.56557	126.67922	112.97292	0.61047	0.33100

Table 2: Posterior mean of the unknown parameters

Finally, Figure 8 shows the original data x and 1000 samples calculated with the parameters previously computed $\tilde{x}$ (Table 2). The model fits what was expected from the original data.

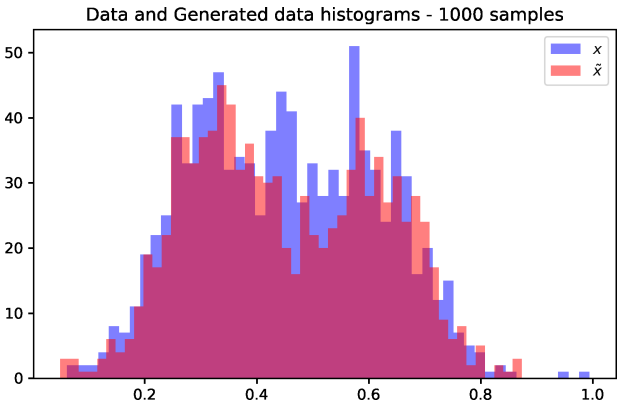


Figure 8: Comparison between both distributions: x: Blue and $\tilde{x}$: Red

One can observe a very good similarity between both histograms (red and blue). To obtain an even better match, a third gaussian could be added to capture the "blue spike" around 0.45.