Probabilistic Unsupervised Learning
Exercise 3: Gaussian Mixture Model

Submission deadline: Monday, December 6, 2021 at 10:00 a.m.

In the programming exercises, support is only provided for Matlab or Python source code.

1. Consider the fruit factory of Fig. 1A. There are blueberries dropping from the ceiling. With probability π_1 a blueberry drops from hole $c = 1$ and with probability π_2 the blueberry drops from hole $c = 2$. As a blueberry either drops from hole 1 or hole 2, the probabilities π_1 and π_2 sum to one. Because of noise (e.g., wind) all blueberries that drop from a specific hole are distributed according to a Gaussian distribution: the blueberries from hole $c = 1$ are distributed according to the distribution $p(x|c = 1, \Theta)$ and the blueberries from hole $c = 2$ according to $p(x|c = 2, \Theta)$. N blueberries drop from the ceiling in this way.

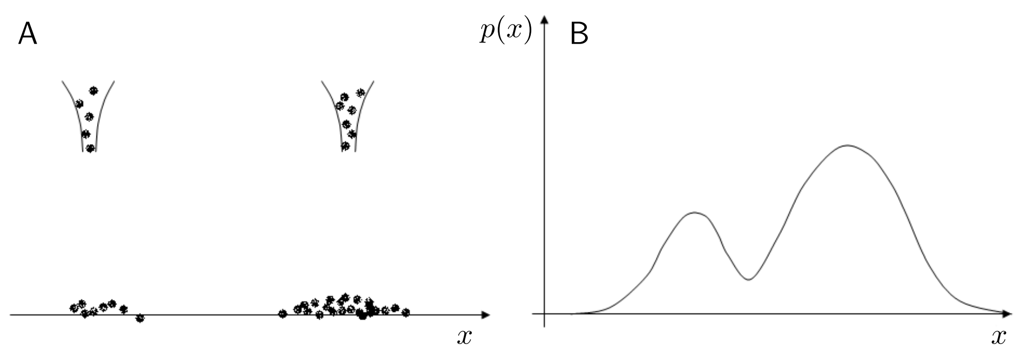

Figure 1: *Blueberries dropping from the ceiling*

The data created by this process are the N real numbers of the positions of the N blueberries. A mathematical model of the data generation processes is given by:

$$p(c|\Theta) = \pi_c \tag{1}$$

$$p(x|c, \Theta) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{1}{2}\frac{(x - \mu_c)^2}{\sigma_c^2}\right) \tag{2}$$

where $c \in \{1, 2\}$, and the parameters Θ are given by $\Theta = (\pi_1, \mu_1, \sigma_1^2, \pi_2, \mu_2, \sigma_2^2)$, with $\pi_1, \pi_2 \in [0, 1]$, $\mu_1, \mu_2 \in \mathbb{R}$ and $\sigma_1^2, \sigma_2^2 \in \mathbb{R}^+$. Note that π_1 and π_2 are not independent as $\pi_1 + \pi_2 = 1$.

The equations (1) and (2) model the *generation* of the N data points. The two equations will therefore be referred to as *generative model*.

[2 *points*] Task A:
Derive the probability density function $p(x|\Theta)$ (see Fig. 1B) for this model.

[1 *point*] Task B:
Using the values $\pi_1 = 0.4, \pi_2 = 0.6, \mu_1 = -2, \mu_2 = 4, \sigma_1^2 = 1, \sigma_2^2 = 10$ for the parameters, plot the probability density function (pdf) obtained in Task A).

[2 *points*] Task C:
Generate 1000 data points using the generative model described above (equations 1 and 2) and the parameters given in Task B). Plot the data points using a one-dimensional plot (scatter-plot).

[2 *points*] Task D:
Using the data points generated in Task C), divide the x-dimension into fifty equally sized bins and compute a histogram on the interval $[-10, 20]$.

[1 *point*] Task E:
Normalize the heights of the bars in Task D) such that they represent an approximation to the probability density in the interval $[-10, 20]$.

[1 *point*] Task F:
Re-plot the pdf used in Task B) together with the normalized histogram obtained in Task E) in the same figure. Try out different numbers of bins and data points. Does the density estimation in Task E) approximate the true underlying density $p(x|\Theta)$ in Task B) equally well for all numbers of bins and data points? Why (not)?

[2 *points*] Task G:
Compute the log-likelihood for N data points under the generative model given in equations 1 and 2.

[1 *point*] Task H:
Evaluate the log-likelihood computed in Task G) using the 1000 data points you have generated in Task C). What is the log-likelihood for other parameters? What do you observe for more and for less data points?

2. Consider the general form of a Gaussian Mixture Model (GMM):

$$p(c|\Theta) = \pi_c \quad \text{with} \quad (c = 1, ..., C); \quad \sum_{c=1}^{C} \pi_c = 1 \tag{3}$$

$$p(\vec{x}|c,\Theta) = \mathcal{N}(\vec{x}; \vec{\mu_c}, \Sigma_c) = \frac{1}{\sqrt{det(2\pi\Sigma_c)}} \exp(-\frac{1}{2}(\vec{x} - \vec{\mu_c})^T \Sigma_c^{-1} (\vec{x} - \vec{\mu_c})) \tag{4}$$

Given N data points we want to find the parameters Θ^* that maximize the log-likelihood $\mathcal{L}(\Theta)$ for the given MoG model.

Use the Expectation Maximization approach to derive the update rules for:

$[4\,points]$ Task A:
$\vec{\mu_c}$ with $c = 1, ..., C$.

$[4\,points]$ Task B:
π_c with $c = 1, ..., C$.

The following hints might be useful:

- Be X a squared matrix, and $x\prime$ a real variable. Then:

$$\frac{d}{dx\prime} \log\left(\det(X)\right) = tr\left(X^{-1}\frac{\partial X}{\partial x'}\right)$$

$$\text{where } \left(\frac{\partial X}{\partial x\prime}\right)_{i,j} = \frac{d}{dx\prime} X_{i,j}$$

  Where log is the natural logarithm, $\det(X)$ and $tr(X)$ are the determinant and the trace over the matrix X respectively. X^{-1} is the inverse of X.

- Be $\vec{x}$ and $\vec{s}$ vectors, W a squared matrix. Then :

$$\frac{\partial(\vec{x}-\vec{s})^T}{\partial \vec{s}} W(\vec{x}-\vec{s}) = -2\,W(\vec{x}-\vec{s}), \text{ where } \left(\frac{\partial(\vec{x}-\vec{s})^T}{\partial \vec{s}} W(\vec{x}-\vec{s})\right)_i = \frac{d(\vec{x}-\vec{s})^T}{ds_i} W(\vec{x}-\vec{s})$$

$$\frac{\partial(\vec{x}^T W \vec{x})}{\partial W} = \vec{x}\,\vec{x}^T$$