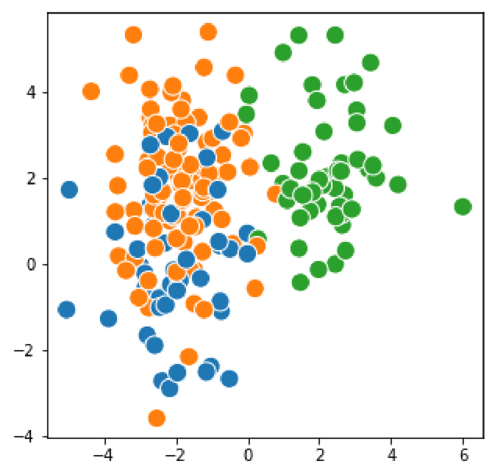
ECE69500: Inference & Learning in Generative Models

Spring 2021

Homework 4

The attached CSV file, `HMMdata.csv`, contains three columns of data: one column of class identities ($\boldsymbol{X}$), and two columns of two-dimensional "features" ($\boldsymbol{Y}$).

1. Plot the data, with each class in a different color, like this:



2. Assume the data $\boldsymbol{Y}$ were generated by a **Gaussian mixture model** with three classes and are observed, while the class identities $\boldsymbol{X}$ are not. In whatever programming language you prefer, implement (from scratch)inference under the GMM,

$$q(\boldsymbol{X}) = \text{Categ}\left[\boldsymbol{\pi}\right]$$
$$q(\boldsymbol{Y}|\boldsymbol{X}) = \mathcal{N}\left(\mathbf{C}\boldsymbol{X}, \boldsymbol{\Sigma}\right)$$

with the parameter values

$$\boldsymbol{\pi} = \left(\frac{3}{16}, \frac{5}{16}, \frac{8}{16}\right)^{\mathrm{T}}, \qquad \mathbf{C} = \begin{pmatrix} -2 & 2 & -2 \\ 0 & 2 & 2 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & -\frac{1}{5} \\ -\frac{1}{5} & 2 \end{pmatrix}$$

Note that we are assuming a fixed covariance matrix across all classes; and that we are treating $\boldsymbol{X}$ as a one-hot vector, so that it selects a column of $\mathbf{C}$ for the mean of the emission.

Thus your code should return a set of posterior probabilities over the latent state $\boldsymbol{X}$: $q(\boldsymbol{X}|\boldsymbol{y})$ for all observations $\boldsymbol{y}$. (As a sanity check, selecting the most probable state will yield an average accuracy of 80%, i.e. `np.mean(np.argmax(qGMM, 0) == X))` is 0.8.)

3. Now assume the data $\boldsymbol{Y}$ were generated by a **hidden Markov model** with three classes and are observed, while again the class identities $\boldsymbol{X}$ are not. In whatever programming language you prefer, implement (from scratch) inference under the HMM,

$$
\begin{aligned}
q(\boldsymbol{X}_1) &= \text{Categ}\left[\boldsymbol{\pi}\right] \\
q(\boldsymbol{X}_n|\boldsymbol{X}_{n-1}) &= \text{Categ}\left[\mathbf{A}\boldsymbol{X}_{n-1}\right] \\
q(\boldsymbol{Y}_n|\boldsymbol{X}_n) &= \mathcal{N}\left(\mathbf{C}\boldsymbol{X}_n, \boldsymbol{\Sigma}\right)
\end{aligned}
$$

with the parameter values

$$
\boldsymbol{\pi} = \left(\frac{3}{5}, \frac{1}{5}, \frac{1}{5}\right), \qquad C = \begin{pmatrix} -2 & 2 & -2 \\ 0 & 2 & 2 \end{pmatrix}, \qquad A = \begin{pmatrix} \frac{4}{10} & \frac{2}{10} & \frac{1}{10} \\ \frac{3}{10} & \frac{5}{10} & \frac{2}{10} \\ \frac{3}{10} & \frac{3}{10} & \frac{7}{10} \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & -\frac{1}{5} \\ -\frac{1}{5} & 2 \end{pmatrix}
$$

(Notice that $\boldsymbol{\pi}$ is different from the GMM.)

Thus your code should return a set of posterior probabilities over the latent state $\boldsymbol{X}_n$. (As a sanity check, selecting the most probable state at each time step from the *filter distribution* will yield an average accuracy of 81.5%; and selecting the most probable state at each time step from the *smoother distribution* will yield an average accuracy of 83%.)