As usual, please hand in on paper form your derivations and answers to the questions. You can use any programming language for your source code (submitted on Studium as per the website instructions). All the requested figures should be printed on paper with clear titles that indicate what the figures represent.

1 Entropy and Mutual Information (18 points)

1. Let X be a discrete random variable on a finite space $\mathcal{X}$ with $|\mathcal{X}| = k$.
   (a) Prove that the entropy $H(X) \geq 0$, with equality only when X is a constant.
   (b) Denote by p the distribution of X and q the uniform distribution on $\mathcal{X}$. What is the relation between the Kullback-Leibler divergence $D(p\|q)$ and the entropy $H(X)$ of the distribution p?
   (c) Deduce an upper bound on the entropy that depends on k.
2. We consider a pair of discrete random variables (X_1, X_2) defined over the finite set $\mathcal{X}_1 \times \mathcal{X}_2$. Let $p_{1,2}$, p_1 and p_2 denote respectively the joint distribution, the marginal distribution of X_1 and the marginal distribution of X_2. The mutual information $I(X_1, X_2)$ is defined as

$$I(X_1, X_2) := \sum_{(x_1,x_2)\in\mathcal{X}_1\times\mathcal{X}_2} p_{1,2}(x_1, x_2) \log \frac{p_{1,2}(x_1, x_2)}{p_1(x_1)p_2(x_2)}.$$

   (a) Prove that $I(X_1, X_2) \geq 0$.
   (b) Show that $I(X_1, X_2)$ can be expressed as a function of $H(X_1), H(X_2)$ and $H(X_1, X_2)$ where $H(X_1, X_2)$ is the entropy of the random variable $X = (X_1, X_2)$.
   (c) What is the joint distribution $p_{1,2}$ of maximal entropy with given marginals p_1 and p_2?

2 HMM – Implementation (82 points)

We consider the same training data as in the previous homework (hwk 3), provided as the `EMGaussian.train` file (and we will test on the corresponding testing data from `EMGaussian.test`), but this time we use an HMM model to account for the possible temporal structure of the data. I.e. we now consider each row of the dataset to be a point $x_t \in \mathbb{R}^2$, where t is the time index (increasing with rows) going from $t = 1, \ldots, T$ rather than thinking of them as *independent* samples as we did in the last homework. The goal of this exercise is to implement the probabilistic inference algorithm (sum-product) on a HMM and its EM algorithm to estimate parameters as well as the Viterbi algorithm to do decoding. It is recommended to make use of the code of the previous homework.

We consider the following HMM model: the chain $(z_t)_{t=1}^T$ has $K = 4$ possible states, with an initial probability distribution $\pi \in \Delta_4$ and a probability transition matrix $A \in \mathbb{R}^{4\times4}$ where

$$A_{ij} = p(z_t = i|z_{t-1} = j),$$

and conditionally on the current state z_t, we have observations obtained from Gaussian emission probabilities $x_t|(z_t = k) \sim \mathcal{N}(x_t|\mu_k, \Sigma_k)$. This is thus a generalization of a GMM with time dependence across the latent states z_t.

1. Implement the α and β-recursions seen in class (and that can be found in chapter 12 of Mike's book with slightly different notation) to compute the *smoothing* distribution $p(z_t|x_1, \ldots, x_T)$ and pair-marginals $p(z_t, z_{t+1}|x_1, \ldots, x_T)$.
   (Recall that $\alpha(z_t) := p(z_t, x_{1:t})$ and $\beta(z_t) := p(x_{(t+1):T}|z_t)$).

2. **(Fake parameters inference).** Consider using the same parameters for the means and covariance matrix of the 4 Gaussians that you should have learned in hwk 3 (with general covariance matrices) with EM. We give them below for your convenience:

$$\mu_1 = \begin{pmatrix} -2.0344 \\ 4.1726 \end{pmatrix} \quad \mu_2 = \begin{pmatrix} 3.9779 \\ 3.7735 \end{pmatrix} \quad \mu_3 = \begin{pmatrix} 3.8007 \\ -3.7972 \end{pmatrix} \quad \mu_4 = \begin{pmatrix} -3.0620 \\ -3.5345 \end{pmatrix}$$

$$\Sigma_1 = \begin{pmatrix} 2.9044 & 0.2066 \\ 0.2066 & 2.7562 \end{pmatrix} \qquad \Sigma_2 = \begin{pmatrix} 0.2104 & 0.2904 \\ 0.2904 & 12.2392 \end{pmatrix}$$

$$\Sigma_3 = \begin{pmatrix} 0.9213 & 0.0574 \\ 0.0574 & 1.8660 \end{pmatrix} \qquad \Sigma_4 = \begin{pmatrix} 6.2414 & 6.0502 \\ 6.0502 & 6.1825 \end{pmatrix}$$

   Using a uniform initial probability distribution $\pi_k = \frac{1}{4}$, and setting A to be the matrix with diagonal coefficients $A_{ii} = \frac{1}{2}$ and off-diagonal coefficients $A_{ij} = \frac{1}{6}$ for all $(i, j) \in \{1, \ldots, 4\}^2$, compute the vectors α_t and β_t for all t on the test data `EMGaussian.test` and compute $p(z_t|x_1, \ldots, x_T)$. Finally, represent $p(z_t|x_1, \ldots, x_T)$ for each of the 4 states as a function of t for the 100 first datapoints in the file. Note that only the 100 time steps should be plotted, but the smoothing is always done with all the data (i.e. $T = 500$). This will be the same for the subsequent questions.
   (In Matlab/Scipy the command `subplot` might be handy to make multiple long horizontal plots.)

3. Derive the M-step update for $\hat{\pi}$, $\hat{A}$, $\hat{\mu}_k$ and $\hat{\Sigma}_k$ (for $k = 1, \ldots, 4$) during the EM algorithm, as a function of the quantities computed during the E step (For the estimate of π, note that we only have *one* long chain here).

4. Implement the EM algorithm to learn the parameters of the model $(\pi, A, \mu_k, \Sigma_k, k = 1 \ldots, 4)$. To make grading easier (avoiding the variability coming from multiple local maxima), use the parameters from question 2 for initialization. Learn the model from the training data in `EMGaussian.train`.

5. Plot the log-likelihood on the train data `EMGaussian.train` and on the test data `EMGaussian.test` as a function of the iterations of the algorithm. Comment.

6. Return in a table the values of the log-likelihoods of the (full-covariance) Gaussian mixture models and of the HMM on the train and on the test data. Compare these values. Does it make sense to make this comparison? Conclude. Compare these log-likelihoods as well with the log-likelihoods obtained for the different models in the previous homework.

7. Provide a description and pseudo-code for the Viterbi decoding algorithm (aka MAP inference algorithm or max-product algorithm) that estimates the most likely sequence of states, i.e. $\arg\max_z p(z_1, \ldots, z_T | x_1, \ldots, x_T)$.

8. Implement Viterbi decoding. For the set of parameters learned with the EM algorithm, compute the most likely sequence of states with the Viterbi algorithm and represent the data in 2D with the cluster centers and with markers of different colors for the datapoints belonging to different classes.

9. For the datapoints in the test file `EMGaussian.test`, compute the marginal probability $p(z_t | x_1, \ldots, x_T)$ for each point to be in state $\{1, 2, 3, 4\}$ for the parameters learned on the training set. For each state plot the probability of being in that state as a function of time for the 100 first points (i.e., as a function of the datapoint index in the file).

10. For each of these same 100 points, compute their most likely state according to the marginal probability computed in the previous question. Make a plot representing the most likely state in $\{1, 2, 3, 4\}$ as function of time for these 100 points.

11. Run Viterbi on the test data. Compare the most likely sequence of states obtained for the 100 first data points with the sequence of states obtained in the previous question. Make a similar plot. Comment.

12. In this problem the number of states K was known. How would you choose the number of states if you did not know it?