Soft Clustering and Embedding

In this problem set we will implement *"soft" K-means* clustering, which is the mean-field approximation of pairwise clustering with squared Euclidean distances. The file `cluster.dat` contains a data set of $p = 500$ (2-dimensional) observations generated from four different Gaussians with four different means.

11.1 Soft K-means Clustering (5 points)

Implement the soft K-means clustering algorithm with squared Euclidean distances. Write a program that contains the following components:

Initialization –

- Set $K = 8$ initial prototypes $\mathbf{w}_q$ randomly around the data set mean
- Choose a convergence criterion γ

Optimization –

1. For fixed β (no annealing), let the optimization procedure run until convergence $\|\mathbf{w}_q^{new} - \mathbf{w}_q^{old}\| < \gamma \ \forall q$. Repeat this for different $\beta \in [0.2, 20]$ e.g. in steps of $\Delta\beta = 0.2$. Use the same initial prototypes for all runs.
2. In additional simulations, run the optimization for $K = 4, 6, 8$ using an annealing schedule: increase β after each iteration. E.g. $\beta_0 = 0.2,\ \tau = 1.1,\ \beta_{t+1} = \tau\beta_t$.

Plotting –

- Visualize the data set, the initial prototypes and the final prototypes for each (fixed) β in one scatter plot.
- Plot the first coordinates of the final prototypes against the β and interpret the results.
- Show the data set, initial and final prototypes of the "annealed" clustering solutions for $K = 4, 6, 8$ in a scatter plot.

11.2 Self-Organizing Maps (5 points)

Self-Organizing Maps (SOM) can be used for dimensionality reduction and clustering. In this exercise we fit a one-dimensional map to a two-dimensional data set.

- Generate $P = 1000$ data points uniformly distributed in the interval $\boldsymbol{x} \in [0, 2] \times [0, 1]$
- Implement a one-dimensional self-organizing map (Kohonen network) using a Gaussian neighborhood function

$$h_{qp} = e^{-\frac{(q-p)^2}{2\sigma^2}}$$

- Fit different maps with $k \in \{4, 8, 16, 32, 64, 128\}$ nodes (prototypes) to the data.

  Note: both learning rate η and the neighborhood width σ should be annealed during learning. It is important that the start value σ_0 is large enough in order to unfold the randomly initialized and thus scrambled map in the first iterations.
- Plot the final map in the data space, i.e. the locations of the prototypes and their connections, for each number of nodes k.
- Plot the mean distance of the best-matching prototype $p_1^{(\alpha)}$ and the second-best-matching prototype $p_2^{(\alpha)}$

$$< d >= \frac{1}{P} \sum_{\alpha} |p_1^{(\alpha)} - p_2^{(\alpha)}|,$$

  where

$$p_1^{(\alpha)} = \operatorname*{argmin}_{r} |\boldsymbol{x}^{(\alpha)} - \boldsymbol{w}_r| \qquad \text{and} \qquad p_2^{(\alpha)} = \operatorname*{argmin}_{r \neq p_1^{(\alpha)}} |\boldsymbol{x}^{(\alpha)} - \boldsymbol{w}_r|$$

  over k (with logarithmic scale for k). Note that the distance d is measured in the "map space", so it equals the number of edges between the two nodes.

Total points: 10