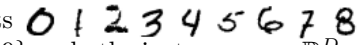
The objective of this lab is for you to explore the behavior of several representative clustering algorithms in Matlab and apply them to some datasets. The TA will first demonstrate the results of the algorithms on a toy dataset and the MNIST dataset. Then, you will replicate those results and further explore the datasets with the algorithms.

We provide you with the following:

- The script `lab04.m` sets up the problem (toy dataset or MNIST) and plots various figures. The actual algorithms are implemented in the functions below.
- `kmeans.m`, `GMEM.m`, `mean_shift.m` and `cc_eball.m`: implement clustering by k-means, EM with Gaussian mixtures, mean-shift and connected-components, respectively.
- For several of the algorithms we use the Gaussian mixture tools.

I Datasets

Construct your own toy datasets in 2D, such as Gaussian clusters with more or less overlap, or clusters with curved shapes as in the 2moons dataset. You will also use the MNIST dataset of handwritten digits  . Since clustering algorithms are unsupervised, they do not use the class labels $y_n \in \{0, \dots, 9\}$, only the instances $\mathbf{x} \in \mathbb{R}^D$ (where $D = 784$). You can use the labels to see if they agree with the resulting clusters found by an algorithm.

II Using clustering algorithms

Algorithms and plots We consider the following algorithms: k-means, EM for Gaussian mixtures with full covariances, mean-shift and connected-components. The figure shows pseudocode for the algorithms. For algorithms that take an infinite number of iterations to converge, we stop them after `maxit` iterations (e.g. 100) or once the error function changes by less than a small value `tol` (e.g. 10^{-3}). With toy datasets in 2D, we plot the following figures:

- For every algorithm: the dataset in 2D, with points colored according to the cluster they belong to.
- For k-means: the value of the error function after each iteration; it should decrease monotonically and stop in a finite number of iterations.
- For EM with Gaussian mixtures: the value of the log-likelihood function after each iteration; it should increase monotonically. To get a hard clustering, we assign point $\mathbf{x}_n$ to cluster k if $p(k|\mathbf{x}_n) > p(j|\mathbf{x}_n)\ \forall j \neq k$. To get a soft clustering (which is more informative), we plot $p(k|\mathbf{x})$ itself for each cluster as a function of $\mathbf{x} \in \mathbb{R}^2$ (as a color plot, or as a contour plot for each cluster).
- For mean-shift: the contours of the kernel density estimate and its modes; and, for any given point $\mathbf{x}_n$, the value of the density $p(\mathbf{x})$ after each mean-shift iteration (initialized from $\mathbf{x}_n$), which should increase monotonically.
- For connected-components: the dataset in 2D, with points connected by edges in the ϵ-ball graph.

With the MNIST dataset, try EM with Gaussian mixtures (also k-means) with different K values and plot:

1. The mean $\boldsymbol{\mu}_k$ of each cluster $k = 1, \dots, K$, as a grayscale image, with its mixing proportion $\pi_k = p(k)$.
2. The posterior probabilities $p(k|\mathbf{x}_n)$ for $k = 1, \dots, K$ for a given digit image $\mathbf{x}_n$, plotted as a bar chart.

Exploration Explore each algorithm in different settings. First, using the same dataset:

- Try different values of the user parameter (number of clusters K for k-means and Gaussian mixtures with EM, bandwidth $\sigma > 0$ for mean-shift, scale $\epsilon > 0$ for connected-components).
- For algorithms that depend on the initialization (k-means and EM), try different random initializations.

Then, explore the algorithms and plots with different datasets, number of clusters, clusters with more or less overlap, with different shapes, etc. See the end of file `lab04.m` for suggestions of things to explore.

Notes

- Some of these algorithms may be slow (in particular, EM and mean-shift). Use small datasets to get results fast.
- For EM with Gaussian mixtures, we add a small number to the diagonal of each covariance matrix $\mathbf{\Sigma}_k$ (e.g. $10^{-10}\,\mathrm{tr}\,(\mathbf{\Sigma}_k)\,/D$) to make $\mathbf{\Sigma}_k$ be full rank. We do this right after updating $\mathbf{\Sigma}_k$ in the M step.

k-MEANS ALGORITHM

$\{\boldsymbol{\mu}_k\}_{k=1}^K \leftarrow K$ random points from $\{\mathbf{x}_n\}_{n=1}^N$
repeat
  for $n \in \{1,\ldots,N\}$
    $k^* = \arg\min_{k=1,\ldots,K} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|$ — closest mean to $\mathbf{x}_n$
    $z_{nk^*} = 1$ and $z_{nk} = 0\ \forall k \neq k^*$
  for $k \in \{1,\ldots,K\}$
    $\boldsymbol{\mu}_k \leftarrow \sum_{n=1}^N z_{nk}\mathbf{x}_n / \sum_{n=1}^N z_{nk}$ — mean of points in cluster k
until stop
return $\{\boldsymbol{\mu}_k\}_{k=1}^K$, $\mathbf{Z}$

GAUSSIAN MIXTURE ESTIMATED WITH EM ALGORITHM

Initialize $\{\pi_k, \boldsymbol{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^K$ from k-means
repeat
  for $n \in \{1,\ldots,N\}$
    $z_{nk} \leftarrow p(k|\mathbf{x}_n) = \ldots$ — E step
  for $k \in \{1,\ldots,K\}$
    $\pi_k \leftarrow \ldots$, $\boldsymbol{\mu}_k \leftarrow \ldots$, $\mathbf{\Sigma}_k \leftarrow \ldots$ — M step
until stop
return $\{\pi_k, \boldsymbol{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^K$

GAUSSIAN MEAN-SHIFT ALGORITHM

for $n \in \{1,\ldots,N\}$
  $\mathbf{x} \leftarrow \mathbf{x}_n$
  repeat
    $\forall n$: $p(n|\mathbf{x}) \leftarrow \frac{\exp\left(-\frac{1}{2}\|(\mathbf{x}-\mathbf{x}_n)/\sigma\|^2\right)}{\sum_{n'=1}^N \exp\left(-\frac{1}{2}\|(\mathbf{x}-\mathbf{x}_{n'})/\sigma\|^2\right)}$
    $\mathbf{x} \leftarrow \sum_{n=1}^N p(n|\mathbf{x})\mathbf{x}_n$
  until stop
  $\mathbf{z}_n \leftarrow \mathbf{x}$ — mode found from $\mathbf{x}_n$
end
return connected-components($\{\mathbf{z}_n\}_{n=1}^N$,ϵ) — aggregate modes found

CONNECTED-COMPONENTS ALGORITHM (ϵ-BALL GRAPH)

Define an ϵ-ball graph:

- vertices $\mathbf{x}_1,\ldots,\mathbf{x}_N$
- edges $(\mathbf{x}_n,\mathbf{x}_m) \Leftrightarrow d(\mathbf{x}_n,\mathbf{x}_m) < \epsilon$, $\forall n,m = 1,\ldots,N$.

Apply DFS to this graph.
return its connected components

Figure 1: Pseudocode for k-means, EM for Gaussian mixtures, mean-shift for the Gaussian kernel and connected-components for an ϵ-ball graph (using a distance function $d(\cdot,\cdot)$). In all cases, the input is a dataset $\mathbf{x}_1,\ldots,\mathbf{x}_N \in \mathbb{R}^D$ and a user parameter: number of clusters K for k-means and Gaussian mixtures with EM, bandwidth $\sigma > 0$ for mean-shift, and scale $\epsilon > 0$ for connected-components. For details omitted $(\ldots)$, see the lecture notes