10. Clustering
Motivation

• Supervised learning is good for interaction with humans, but labels from a supervisor are sometimes hard to obtain

• Clustering is \textit{unsupervised} learning, i.e. it tries to learn only from the data

• Main idea: find a similarity measure and group similar data objects together

• Clustering is a very old research field, many approaches have been suggested

• Main problem in most methods: how to find a good number of clusters
In unsupervised learning, there is no *ground truth* information given. Most Unsupervised Learning methods are based on Clustering.
K-means Clustering

- Given: data set \( \{x_1, \ldots, x_N\} \), number of clusters \( K \)
- Goal: find cluster centers \( \{\mu_1, \ldots, \mu_K\} \) so that

\[
J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2
\]

is minimal, where \( r_{nk} = 1 \) if \( x_n \) is assigned to \( \mu_k \)

- Idea: compute \( r_{nk} \) and \( \mu_k \) iteratively
- Start with some values for the cluster centers
- Find optimal assignments \( r_{nk} \)
- Update cluster centers using these assignments
- Repeat until assignments or centers don’t change
K-means Clustering

Initialize cluster means: \( \{ \mu_1, \ldots, \mu_K \} \)
K-means Clustering

Find optimal assignments:

\[ r_{nk} = \begin{cases} 
1 & \text{if } k = \arg \min_j \| x_n - \mu_j \| \\
0 & \text{otherwise}
\end{cases} \]
K-means Clustering

Find new optimal means:

\[
\frac{\partial J}{\partial \mu_k} = 2 \sum_{n=1}^{N} r_{nk} (x_n - \mu_k) = 0
\]

\[
\Rightarrow \mu_k = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}
\]
K-means Clustering

Find new optimal assignments:

\[ r_{nk} = \begin{cases} 
1 & \text{if } k = \arg \min_j \|x_n - \mu_j\| \\
0 & \text{otherwise}
\end{cases} \]
K-means Clustering

Iterate these steps until means and assignments do not change any more
2D Example

- Real data set
- Random initialization
- Magenta line is “decision boundary”
• After every step the cost function $J$ is minimized
• Blue steps: update assignments
• Red steps: update means
• Convergence after 4 rounds
K-means for Segmentation

$K = 2$

$K = 3$

$K = 10$

Original image
K-Means: Additional Remarks

• K-means converges always, but the minimum is not guaranteed to be a global one

• There is an **online** version of \( K \)-means
  
  • After each addition of \( x_n \), the nearest center \( \mu_k \) is updated:
  
  \[
  \mu_{k}^{\text{new}} = \mu_{k}^{\text{old}} + \eta_n (x_n - \mu_{k}^{\text{old}})
  \]

• The **\( K \)-medoid** variant:
  
  • Replace the Euclidean distance by a general measure \( V \).

\[
\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} V(x_n, \mu_k)
\]
Mixtures of Gaussians

• Assume that the data consists of $K$ clusters
• The data within each cluster is Gaussian
• For any data point $x$ we introduce a $K$-dimensional binary random variable $z$ so that:

$$p(x) = \sum_{k=1}^{K} p(z_k = 1) \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

where

$$z_k \in \{0, 1\}, \quad \sum_{k=1}^{K} z_k = 1$$
A Simple Example

- Mixture of three Gaussians with mixing coefficients
- Left: all three Gaussians as contour plot
- Right: samples from the mixture model, the red component has the most samples
Parameter Estimation

- From a given set of training data \( \{x_1, \ldots, x_N\} \) we want to find parameters \( (\pi_1, \ldots, K, \mu_1, \ldots, K, \Sigma_1, \ldots, K) \) so that the likelihood is maximized (MLE):

\[
p(x_1, \ldots, x_N \mid \pi_1, \ldots, K, \mu_1, \ldots, K, \Sigma_1, \ldots, K) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)
\]

or, applying the logarithm:

\[
\log p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)
\]

- However: this is not as easy as maximum-likelihood for single Gaussians!
Problems with MLE for Gaussian Mixtures

• Assume that for one $k$ the mean $\mu_k$ is exactly at a data point $x_n$
  • For simplicity: assume that $\Sigma_k = \sigma_k^2 I$
  • Then:
    \[
    \mathcal{N}(x_n | x_n, \sigma_k^2 I) = \frac{1}{\sqrt{2\pi\sigma_k^D}}
    \]
  • This means that the overall log-likelihood can be maximized arbitrarily by letting $\sigma_k \rightarrow 0$ (overfitting)

• Another problem is the identifiability:
  • The order of the Gaussians is not fixed, therefore:
  • There are $K!$ equivalent solutions to the MLE problem
Overfitting with MLE for Gaussian Mixtures

- One Gaussian fits exactly to one data point
- It has a very small variance, i.e. contributes strongly to the overall likelihood
- In standard MLE, there is no way to avoid this!
Expectation-Maximization

- EM is an elegant and powerful method for MLE problems with latent variables.
- Main idea: model parameters and latent variables are estimated iteratively, where average over the latent variables (expectation).
- A typical example application of EM is the Gaussian Mixture model (GMM).
- However, EM has many other applications.
- First, we consider EM for GMMs.
Expectation-Maximization for GMM

• First, we define the responsibilities:

\[ \gamma(z_{nk}) = p(z_{nk} = 1 \mid x_n) \quad z_{nk} \in \{0, 1\} \]

\[ \sum_k z_{nk} = 1 \]
Expectation-Maximization for GMM

• First, we define the responsibilities:

\[
\gamma(z_{nk}) = p(z_{nk} = 1 \mid x_n)
\]

\[
= \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)}
\]
Expectation-Maximization for GMM

• First, we define the responsibilities:

\[
\gamma(z_{nk}) = p(z_{nk} = 1 \mid x_n) = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)}
\]

• Next, we derive the log-likelihood wrt. to \(\mu_k\):

\[
\frac{\partial \log p(X \mid \pi, \mu, \Sigma)}{\partial \mu_k} \equiv 0
\]
**Expectation-Maximization for GMM**

- First, we define the responsibilities:

\[ \gamma(z_{nk}) = p(z_{nk} = 1 \mid x_n) \]

\[ = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} \]

- Next, we derive the log-likelihood wrt. to \( \mu_k \):

\[ \frac{\partial \log p(X \mid \pi, \mu, \Sigma)}{\partial \mu_k} \equiv 0 \]

and we obtain:

\[ \mu_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \]
Expectation-Maximization for GMM

• We can do the same for the covariances:

\[
\frac{\partial \log p(X \mid \pi, \mu, \Sigma)}{\partial \Sigma_k} \stackrel{!}{=} 0
\]

and we obtain:

\[
\Sigma_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})}
\]

• Finally, we derive wrt. the mixing coefficients \( \pi_k \):

\[
\frac{\partial \log p(X \mid \pi, \mu, \Sigma)}{\partial \pi_k} \stackrel{!}{=} 0 \quad \text{where:} \quad \sum_{k=1}^{K} \pi_k = 1
\]
Expectation-Maximization for GMM

• We can do the same for the covariances:

\[
\frac{\partial \log p(X \mid \pi, \mu, \Sigma)}{\partial \Sigma_k} \stackrel{!}{=} 0
\]

and we obtain:

\[
\Sigma_k = \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T \frac{1}{\sum_{n=1}^{N} \gamma(z_{nk})}
\]

• Finally, we derive wrt. the mixing coefficients \( \pi_k \):

\[
\frac{\partial \log p(X \mid \pi, \mu, \Sigma)}{\partial \pi_k} \stackrel{!}{=} 0 \quad \text{where:} \quad \sum_{k=1}^{K} \pi_k = 1
\]

and the result is:

\[
\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})
\]
Algorithm Summary

1. Initialize means $\mu_k$, covariance matrices $\Sigma_k$, and mixing coefficients $\pi_k$.
2. Compute the initial log-likelihood $\log p(X \mid \pi, \mu, \Sigma)$.
3. **E-Step.** Compute the responsibilities:
   \[
   \gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)}
   \]
4. **M-Step.** Update the parameters:
   \[
   \mu_{k}^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \quad \Sigma_{k}^{\text{new}} = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_{k}^{\text{new}})(x_n - \mu_{k}^{\text{new}})^T}{\sum_{n=1}^{N} \gamma(z_{nk})} \quad \pi_{k}^{\text{new}} = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})
   \]
5. Compute log-likelihood; if not converged go to 3.
The Same Example Again

\begin{align*}
\text{(a)} & \quad L = 1 \\
\text{(b)} & \quad L = 2 \\
\text{(c)} & \quad L = 5 \\
\text{(d)} & \quad L = 10 \\
\text{(e)} & \quad L = 20
\end{align*}
Why is it Called “EM”?

Assume for a moment that we observe $X$ and the binary latent variables $Z$. The likelihood is then:

$$p(X, Z \mid \pi, \mu, \Sigma) = \prod_{n=1}^{N} p(z_n \mid \pi)p(x_n \mid z_n, \mu, \Sigma)$$

where

$$p(z_n \mid \pi) = \prod_{k=1}^{K} \pi^z_{nk}$$

and

$$p(x_n \mid z_n, \mu, \Sigma) = \prod_{k=1}^{K} \mathcal{N}(x_n \mid \mu_k, \Sigma_k)^{z_{nk}}$$

which leads to the log-formulation:

$$\log p(X, Z \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk}(\log \pi_k + \log \mathcal{N}(x_n \mid \mu_k, \Sigma_k))$$
Why is it Called “EM”?

Instead of maximizing the joint log-likelihood, we maximize its expectation under the latent variable distribution:

$$
\mathbb{E}_Z[\log p(X, Z | \pi, \mu, \Sigma)] = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_Z[z_{nk}](\log \pi_k + \log \mathcal{N}(x_n | \mu_k, \Sigma_k))
$$
Why is it Called “EM”?

Instead of maximizing the joint log-likelihood, we maximize its **expectation** under the latent variable distribution:

\[ \mathbb{E}_Z[\log p(X, Z \mid \pi, \mu, \Sigma)] = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_Z[z_{nk}](\log \pi_k + \log \mathcal{N}(x_n \mid \mu_k, \Sigma_k)) \]

where the latent variable distribution per point is:

\[ p(z_n \mid x_n, \theta) = \frac{p(x_n \mid z_n, \theta)p(z_n \mid \theta)}{p(x_n \mid \theta)} \quad \theta = (\pi, \mu, \Sigma) \]

\[ = \frac{\prod_{l=1}^{K} (\pi_l \mathcal{N}(x_n \mid \mu_l, \Sigma_l))^{z_{nl}}}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} \]
Observations

• Compared to K-means, points can now belong to both clusters (soft assignment)
• In addition to the cluster center, a covariance is estimated by EM
• Initialization is the same as used for K-means
• Number of iterations needed for EM is much higher
• Also: each cycle requires much more computation
• Therefore: start with K-means and run EM on the result of K-means (covariances can be initialized to the sample covariances of K-means)
• EM only finds a local maximum of the likelihood!
Clustering using Mixture Models

The full posterior of the Gaussian Mixture Model is

\[ p(X, Z, \mu, \Sigma, \pi) = p(X \mid Z, \mu, \Sigma)p(Z \mid \pi)p(\pi \mid \alpha)p(\mu, \Sigma \mid \lambda) \]

- data likelihood (Gaussian)
- correspondence prob. (Multinomial)
- mixture prior (Dirichlet)
- parameter prior (Gauss-IW)

In this model, we use:

- \( \mu = (\mu_1, \ldots, \mu_K) \)
- \( \Sigma = (\Sigma_1, \ldots, \Sigma_K) \)
- \( (\mu_k, \Sigma_k) = \theta_k \)
Clustering using Mixture Models

The full posterior of the Gaussian Mixture Model is

\[ p(X, Z, \mu, \Sigma, \pi) = p(X \mid Z, \mu, \Sigma)p(Z \mid \pi)p(\pi \mid \alpha)p(\mu, \Sigma \mid \lambda) \]

- **Data likelihood (Gaussian)**
- **Correspondence prob. (Multinomial)**
- **Mixture prior (Dirichlet)**
- **Parameter prior (Gauss-IW)**

Given this model, we can create new samples:
1. Sample \( \pi, \theta_k \) from priors
2. Sample correspondence \( z_i \)
3. Sample data point \( x_i \)
Clustering using Mixture Models

The full posterior of the Gaussian Mixture Model is

\[ p(X, Z, \mu, \Sigma, \pi) = p(X | Z, \mu, \Sigma)p(Z | \pi)p(\pi | \alpha)p(\mu, \Sigma | \lambda) \]

An equivalent formulation of this model is this:
1. Sample \( \pi, \theta_k \) from priors
2. Sample params \( \bar{\theta}_i \) from:
   \[ p(\bar{\theta}_i | \pi, \theta_k) = \sum_{k=1}^{K} \pi_k \delta(\theta_k, \bar{\theta}_i) \]
3. Sample data point \( x_i \)
Clustering using Mixture Models

What is the difference in that model?

- there is one parameter $\bar{\theta}_i$ for each observation $x_i$
- intuitively: we first sample the location of the cluster and then the data that corresponds to it

In general, we use the notation:

$$
\begin{align*}
\pi & \sim \text{Dir}\left(\frac{\alpha}{K}1\right) \\
\theta_k & \sim H(\lambda) \quad \text{“Base distribution”} \\
\bar{\theta}_i & \sim G(\pi, \theta_k) \quad \text{where} \\
G(\pi, \theta_k) &= \sum_{k=1}^{K} \pi_k \delta(\theta_k, \bar{\theta}_i)
\end{align*}
$$

However: We need to know $K$