DDA3020 Machine Learning: Lecture 15 Gaussian Mixture Models

Baoyuan Wu
School of Data Science, CUHK-SZ

April 22/24, 2024

Mixture Models

- We model the joint distribution over $(\mathbf{x}, z)$ as follows

$$p(\mathbf{x}, z) = p(\mathbf{x}|z)p(z),$$

  where $\mathbf{x}$ denotes a feature variable, and z denotes the class label variable.
- However, we do not have the class labels z in unsupervised clustering.
- In this case, we can model the marginal distribution over $\mathbf{x}$ as follows

$$p(\mathbf{x}) = \sum_z p(\mathbf{x}, z) = \sum_z p(\mathbf{x}|z)p(z).$$

- This is called as mixture models.

Gaussian Mixture Model (GMM)

The most common mixture model is called as Gaussian mixture model (GMM).

- A GMM represents a **distribution** as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right),$$

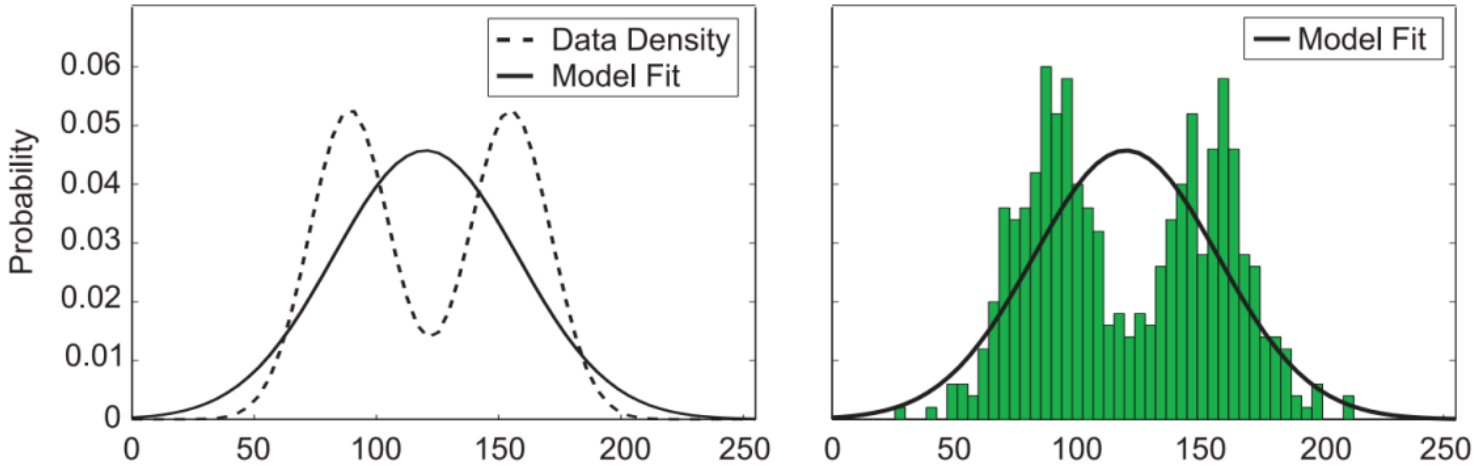
  with π_k the mixing coefficients, where: $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k \geq 0, \forall k$. And,

$$\mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right),$$

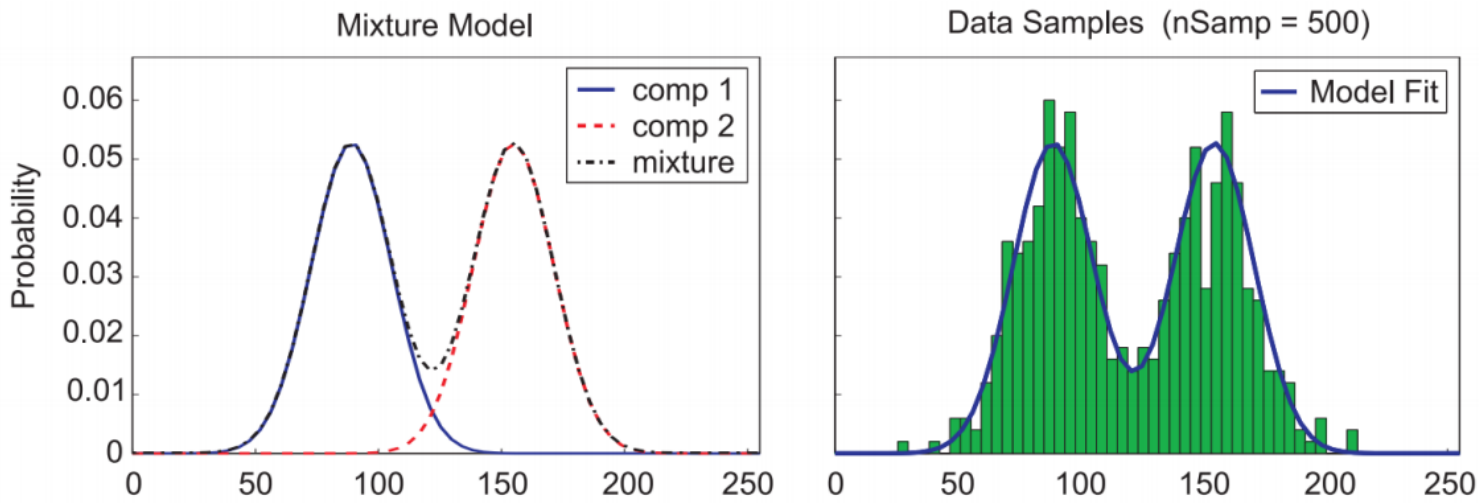
  with $|\boldsymbol{\Sigma}_k| = \det(\boldsymbol{\Sigma}_k)$ denotes the determinant of $\boldsymbol{\Sigma}_k$, d indicates the dimension of $\mathbf{x}$.
- GMM is a density estimator. If given enough Gaussian components, GMM is universal approximator of densities.
- In general, mixture models are very powerful, but difficult to optimize.

Visualizing a Mixture of Gaussians – 1D Gaussians

- If you fit a Gaussian to data:

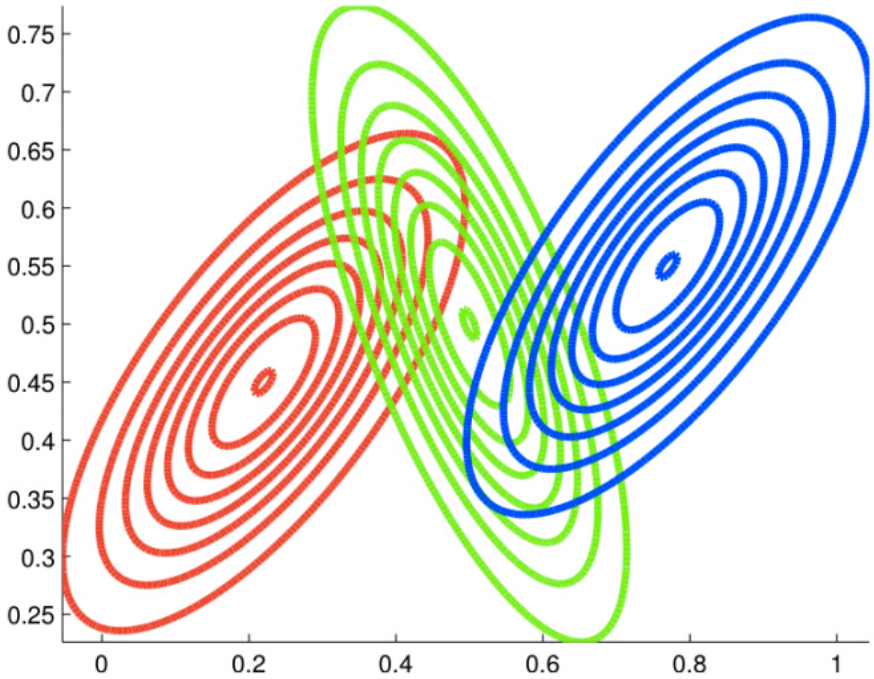


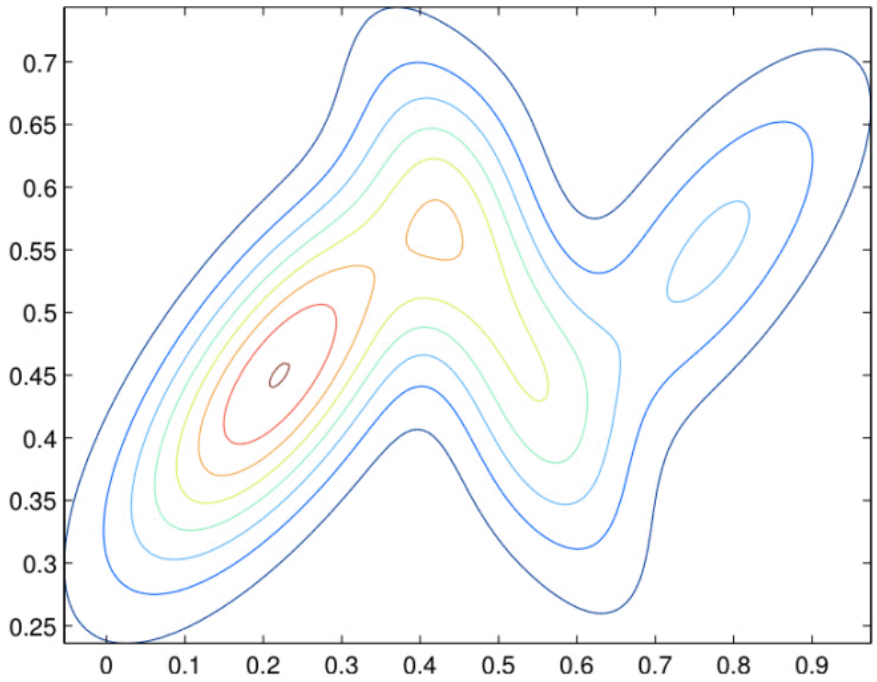
- Now, we are trying to fit a GMM (with $K = 2$ in this example):

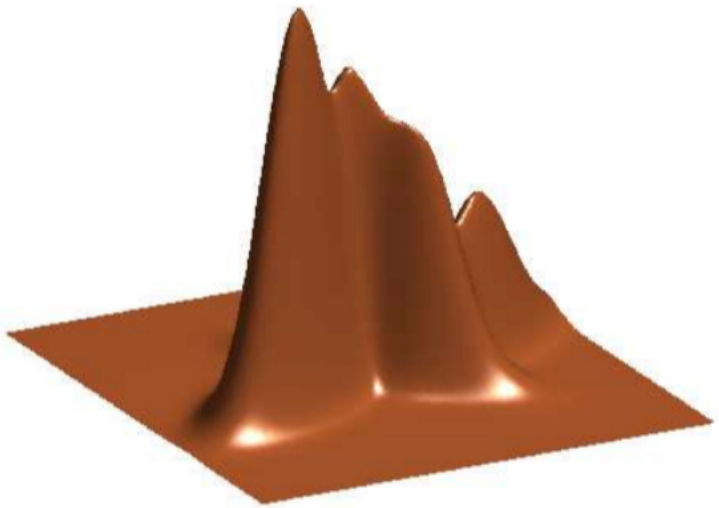


[Slide credit: K. Kutulakos]

Visualizing a Mixture of Gaussians – 2D Gaussians







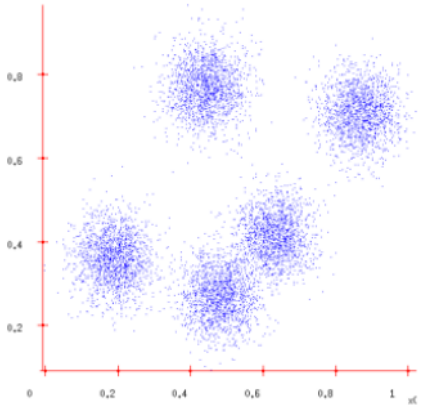
Fitting GMMs: Maximum Likelihood

- The log likelihood is

$$\log \mathcal{L}(\boldsymbol{\Theta}) = \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \right) \right),$$

  where $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$, $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$, and $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K\}$.
- We aim to learn the parameters $\boldsymbol{\Theta}$ by maximizing the above log likelihood.
- Due to the log-sum-exp operation, we cannot obtain a closed-form solution by setting the derivative to zero.
- Of course you can choose gradient based method. However, in the following we will introduce a more elegant optimization method.

Latent Variable



- We introduce a hidden (latent) variable z, indicating which Gaussian component generates the observation $\mathbf{x}$, with some probability.
- Let $z \sim \text{Categorical}\ (\boldsymbol{\pi})$, where $\boldsymbol{\pi} \geq 0, \quad \sum_k \pi_k = 1$.
- Then:

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}, z = k) = \sum_{k=1}^{K} \underbrace{p(z = k)}_{\pi_k} \underbrace{p(\mathbf{x} \mid z = k)}_{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}.$$

- This breaks a complicated distribution into simple components - the price is the hidden variable.

Latent Variable Models

Latent variable model (LVM):

- Definition: A latent variable model is a statistical model that relates a set of observable variables to a set of latent variables.
- Some model variables may be unobserved, either at training or at testing time, or both. Variables which are always unobserved are called **latent variables**, or sometimes **hidden variables**.
- We may want to intentionally introduce latent variables to model complex dependencies between variables – this can actually simplify the model. As shown in the above example, the complex distribution $p(\mathbf{x})$ can be represented by the weighted summation of several simple distributions $p(\mathbf{x} \mid z = k)$.
- According to the type of latent variables, there are two types of LVMs,
  - LVM with continuous latent variables, *e.g.*, factor analysis
  - LVM with discrete latent variables, *e.g.*, mixture models

Reference:

https://en.wikipedia.org/wiki/Latent_variable_model

Back to GMM

- A Gaussian mixture distribution:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right).$$

- We had: $z \sim$ Categorical $(\boldsymbol{\pi})$, *i.e.*, $p(z = k \mid \boldsymbol{\pi}) = \pi_k$, where $\boldsymbol{\pi} \geq 0$, $\sum_k \pi_k = 1$.
- Joint distribution: $p(\mathbf{x}, z) = p(z)p(\mathbf{x} \mid z)$
- Log-likelihood:

$$\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln p\left(\mathbf{x}^{(n)} \mid \pi, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} p\left(\mathbf{x}^{(n)}, z^{(n)} = k \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$= \sum_{n=1}^{N} \ln \sum_{k=1}^{K} p\left(\mathbf{x}^{(n)} \mid z^{(n)} = k; \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) p(z^{(n)} = k \mid \boldsymbol{\pi}).$$

- Note: we have a hidden variable $z^{(n)}$ for every observation $\mathbf{x}^{(n)}$
- How can we optimize this problem, since there is sum inside the log?

Maximum Likelihood

- If we knew $z^{(n)}$ for every $\mathbf{x}^{(n)}$, the maximum log likelihood problem is easy:

$$\max \ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} 1_{\left[z^{(n)}=k\right]} \cdot p\left(\mathbf{x}^{(n)}, z^{(n)} = k \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

$$= \sum_{n=1}^{N} \left[\ln\left(1_{\left[z^{(n)}=k\right]} \cdot p\big(\mathbf{x}^{(n)} \mid z^{(n)} = k; \boldsymbol{\mu}, \boldsymbol{\Sigma}\big)\right) + \ln\left(1_{\left[z^{(n)}=k\right]} \cdot p\big(z^{(n)} = k \mid \boldsymbol{\pi}\big)\right)\right]$$

  with the constraint $1 - \sum_{k=1}^{K} \pi_k = 0$.

- For the above constrained optimization problem, we also resort to KKT conditions based on Lagrangian function, as follows:

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda) = -\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda(1 - \sum_{k=1}^{K} \pi_k).$$

  Note that the original primal problem is maximizing $\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, thus it should be $-\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ in the Lagrangian function.

Maximum Likelihood

- Lagrangian function:

$$\mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda) = -\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda(1 - \sum_{k=1}^{K} \pi_k).$$

- Since there is only one equality constraint, there are only stationary and primal feasibility constraints in KKT conditions. Specifically, $\forall k$, we have

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda)}{\partial \boldsymbol{\mu}_k} = \frac{-\partial \sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \ln p(\mathbf{x}^{(n)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\mu}_k} = \mathbf{0}, \tag{1}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda)}{\partial \boldsymbol{\Sigma}_k} = \frac{-\partial \sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \ln p(\mathbf{x}^{(n)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0}, \tag{2}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda)}{\partial \pi_k} = \frac{-\partial \sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \ln \pi_k}{\partial \pi_k} - \lambda = 0, \tag{3}$$

$$1 - \sum_{k=1}^{K} \pi_k = 0. \tag{4}$$

Maximum Likelihood

- The solution to $\boldsymbol{\mu}_k$:

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda)}{\partial \boldsymbol{\mu}_k} = \frac{-\partial \left[\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \ln p(\mathbf{x}^{(n)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]}{\partial \boldsymbol{\mu}_k} = \mathbf{0}, \tag{5}$$

$$\Rightarrow \frac{\partial \left[\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} (-\frac{1}{2})(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)\right]}{\partial \boldsymbol{\mu}_k} = \mathbf{0}, \tag{6}$$

$$\Rightarrow \sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) = \mathbf{0}, \tag{7}$$

$$\Rightarrow \sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k) = \mathbf{0}, \tag{8}$$

$$\Rightarrow \boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \mathbf{x}^{(n)}}{\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}}. \tag{9}$$

Maximum Likelihood

- The solution to $\boldsymbol{\Sigma}_k$:

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda)}{\partial \boldsymbol{\Sigma}_k} = \frac{-\partial \left[\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \ln p(\mathbf{x}^{(n)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0}, \tag{10}$$

$$\Rightarrow \frac{\partial \left[\frac{1}{2}\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \left(\ln |\boldsymbol{\Sigma}_k| + (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)\right)\right]}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0}. \tag{11}$$

- We define $\boldsymbol{\Lambda}_k = \boldsymbol{\Sigma}_k^{-1}$, which is called precision matrix. Then, the above equation is equivalent to:

$$\frac{\partial \left[\frac{1}{2}\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \left(\ln |\boldsymbol{\Lambda}_k^{-1}| + (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)\right)\right]}{\partial \boldsymbol{\Lambda}_k} = \mathbf{0}, \tag{12}$$

$$\equiv \frac{\partial \left[\frac{1}{2}\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \left(-\ln |\boldsymbol{\Lambda}_k| + (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k (\mathbf{x}^{(n)} - \boldsymbol{\mu}_k)\right)\right]}{\partial \boldsymbol{\Lambda}_k} = \mathbf{0}, \tag{13}$$

where we utilize that $|\boldsymbol{\Lambda}_k^{-1}| = \frac{1}{|\boldsymbol{\Lambda}_k|}$.

Maximum Likelihood

The solution to $\boldsymbol{\Sigma}_k$:

- Its solution is derived as follows:

$$\frac{\partial \left[\frac{1}{2}\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}\left(-\ln|\boldsymbol{\Lambda}_k| + (\mathbf{x}^{(n)}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Lambda}_k(\mathbf{x}^{(n)}-\boldsymbol{\mu}_k)\right)\right]}{\partial \boldsymbol{\Lambda}_k} = \mathbf{0}, \quad (14)$$

$$\Rightarrow \frac{1}{2}\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}\left(-\boldsymbol{\Lambda}_k^{-1} + (\mathbf{x}^{(n)}-\boldsymbol{\mu}_k)(\mathbf{x}^{(n)}-\boldsymbol{\mu}_k)^\top\right) = \mathbf{0}, \quad (15)$$

$$\Rightarrow \boldsymbol{\Lambda}_k^{-1} = \boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}(\mathbf{x}^{(n)}-\boldsymbol{\mu}_k)(\mathbf{x}^{(n)}-\boldsymbol{\mu}_k)^\top}{\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}}, \quad (16)$$

where we utilize that $\frac{d\ln|\boldsymbol{\Lambda}_k|}{d\boldsymbol{\Lambda}_k} = \boldsymbol{\Lambda}_k^{-1}$.

Reference of matrix derivatives: `https://en.wikipedia.org/wiki/Matrix_calculus`

Maximum Likelihood

- The solution to $\boldsymbol{\pi}$:

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \lambda)}{\partial \pi_k} = \frac{-\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \ln \pi_k}{\partial \pi_k} - \lambda = 0, \tag{17}$$

$$\Rightarrow -\sum_{n=1}^{N} \frac{1_{\left[z^{(n)}=k\right]}}{\pi_k} = \lambda \Rightarrow -\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} = \pi_k \lambda. \tag{18}$$

- Combining with $1-\sum_{k=1}^{K} \pi_k = 0$, we can obtain $\lambda = -N$, which is replaced back to obtain

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}.$$

Maximum Likelihood

Solution summary:

- If we knew $z^{(n)}$ for every $\mathbf{x}^{(n)}$, the maximum log likelihood problem is easy:

$$\max \ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln p\left(\mathbf{x}^{(n)}, z^{(n)} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right),$$

  with the constraint $1 - \sum_{k=1}^{K} \pi_k = 0$.

- Its solution is

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \mathbf{x}^{(n)}}{\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}},$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]} \left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right)\left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right)^T}{\sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}},$$

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} 1_{\left[z^{(n)}=k\right]}.$$

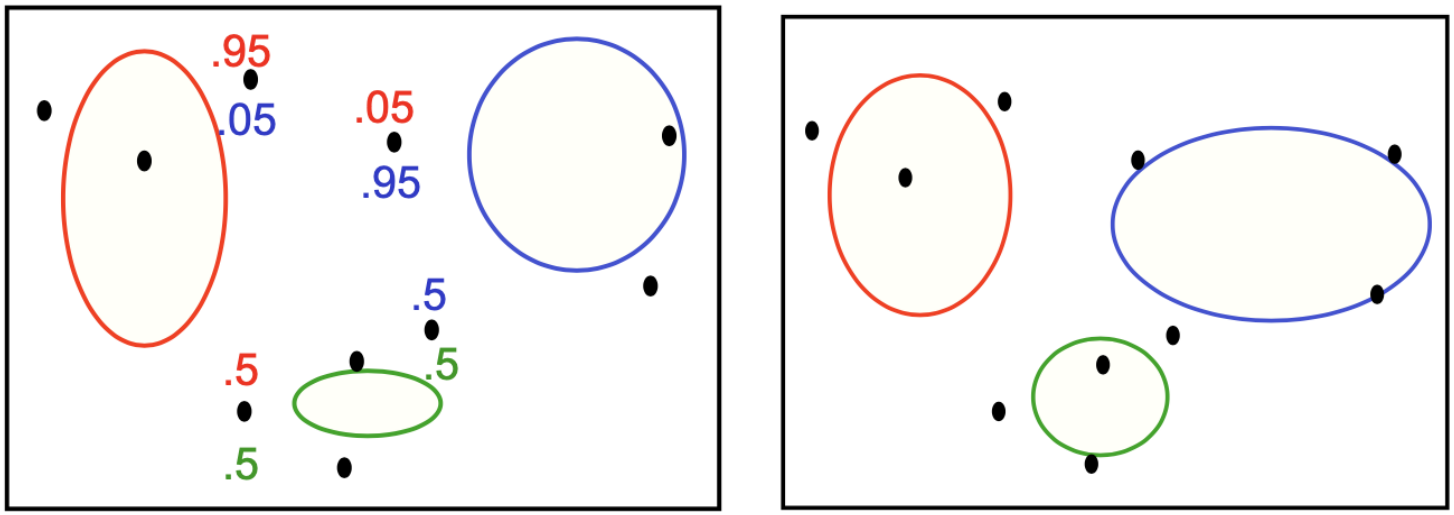
Maximum Likelihood

However, how to learn the parameters when we don't know $z^{(n)}$ for every $\mathbf{x}^{(n)}$?
We have to resort to the expectation maximization algorithm.

Intuitively, How Can We Fit a Mixture of Gaussians?

Optimization uses the **Expectation Maximization algorithm**, which alternates between two steps:

- E-step: Compute the posterior probability over z given the current model, *i.e.*, $p(z|\mathbf{x}; \mathbf{\Theta})$, which tells how much do we think each Gaussian generates each data point.
- M-step: Assuming that the data was really generated this way, update the parameters $\mathbf{\Theta}$ of each Gaussian component, to maximize the probability that it would generate the data it is currently responsible for.



Expectation Maximization for GMM Overview

Elegant and powerful method for finding maximum likelihood solutions for models with latent variables

- **E-step:**
  - In order to adjust the parameters, we must first solve the inference problem: which Gaussian component generated each datapoint?
  - We cannot ensure, so it's a distribution over all possibilities.

$$\gamma_k^{(n)} = p\left(z^{(n)} = k \mid \mathbf{x}^{(n)}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right).$$

- **M-step:**
  - Each Gaussian gets a certain amount of posterior probability for each datapoint.
  - We fit each Gaussian to the weighted datapoints.
  - We can derive closed form updates for all parameters

GMM E-Step: Responsibilities

Lets see how EM works on GMM:

Conditional probability (using Bayes rule) of z given $\mathbf{x}$

$$\begin{aligned}\gamma_k = p(z=k \mid \mathbf{x}) &= \frac{p(z=k)p(\mathbf{x} \mid z=k)}{p(\mathbf{x})} \\ &= \frac{p(z=k)p(\mathbf{x} \mid z=k)}{\sum_{j=1}^{K} p(z=j)p(\mathbf{x} \mid z=j)} \\ &= \frac{\pi_k \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}.\end{aligned}$$

γ_k can be viewed as the **responsibility** of cluster k towards $\mathbf{x}$

GMM E-Step: expected log likelihood

Once we computed $\gamma_k^{(n)} = p\left(z^{(n)} = k \mid \mathbf{x}^{(n)}\right)$, we can compute the expected log likelihood, as follows:

$$
\begin{aligned}
&\sum_n \mathbb{E}_{P(z^{(n)}|\mathbf{x}^{(n)})}\left[\ln\left(P(\mathbf{x}^{(n)}, z^{(n)} \mid \Theta)\right)\right] \\
=&\sum_n \sum_k \gamma_k^{(n)} \left(\ln\left(P(z^{(n)} = k \mid \Theta)\right) + \ln\left(P(\mathbf{x}^{(n)} \mid z^{(n)} = k, \Theta)\right)\right) \\
=&\sum_n \sum_k \gamma_k^{(n)} \left(\ln\left(\pi_k\right) + \ln\left(\mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)\right) \\
=&\sum_n \sum_k \gamma_k^{(n)} \ln\left(\pi_k\right) + \sum_n \sum_k \gamma_k^{(n)} \ln\left(\mathcal{N}(\mathbf{x}^{(n)}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right),
\end{aligned}
$$

where $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}\}$. Note that the above expectation is fully decomposed to each data n and each cluster k, which will facilitate the parameter learning in the following maximization step.

GMM M-Step

Maximization step:

- Given the posterior probability $\gamma_k^{(n)} = p\left(z^{(n)} = k \mid \mathbf{x}^{(n)}\right)$, we want to update the model parameters $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}\}$ by maximizing the expected log likelihood, *i.e.*,

$$\max_{\Theta} \sum_n^N \sum_k^K \gamma_k^{(n)} \ln\left(\pi_k\right) + \sum_n^N \sum_k^K \gamma_k^{(n)} \ln\left(\mathcal{N}(\mathbf{x}^{(n)} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right), \text{ s.t. } \sum_k^K \pi_k = 1.$$

- Following the derivations introduced in previous slides (see page 12-17), it is easy to obtain the following solutions:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma_k^{(n)} \mathbf{x}^{(n)}.$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma_k^{(n)} \left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right)\left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right)^\top.$$

$$\pi_k = \frac{N_k}{N}, \text{ with } N_k = \sum_{n=1}^N \gamma_k^{(n)}.$$

EM for fitting GMM

Summary of EM: **Initialize** the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_k

Iterate until convergence:

- **E-step**: Evaluate the responsibilities given current parameters

$$\gamma_k^{(n)} = p\left(z^{(n)} = k \mid \mathbf{x}^{(n)}; \Theta\right) = \frac{\pi_k \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}.$$

- **M-step**: Re-estimate the parameters given current responsibilities

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \mathbf{x}^{(n)},$$

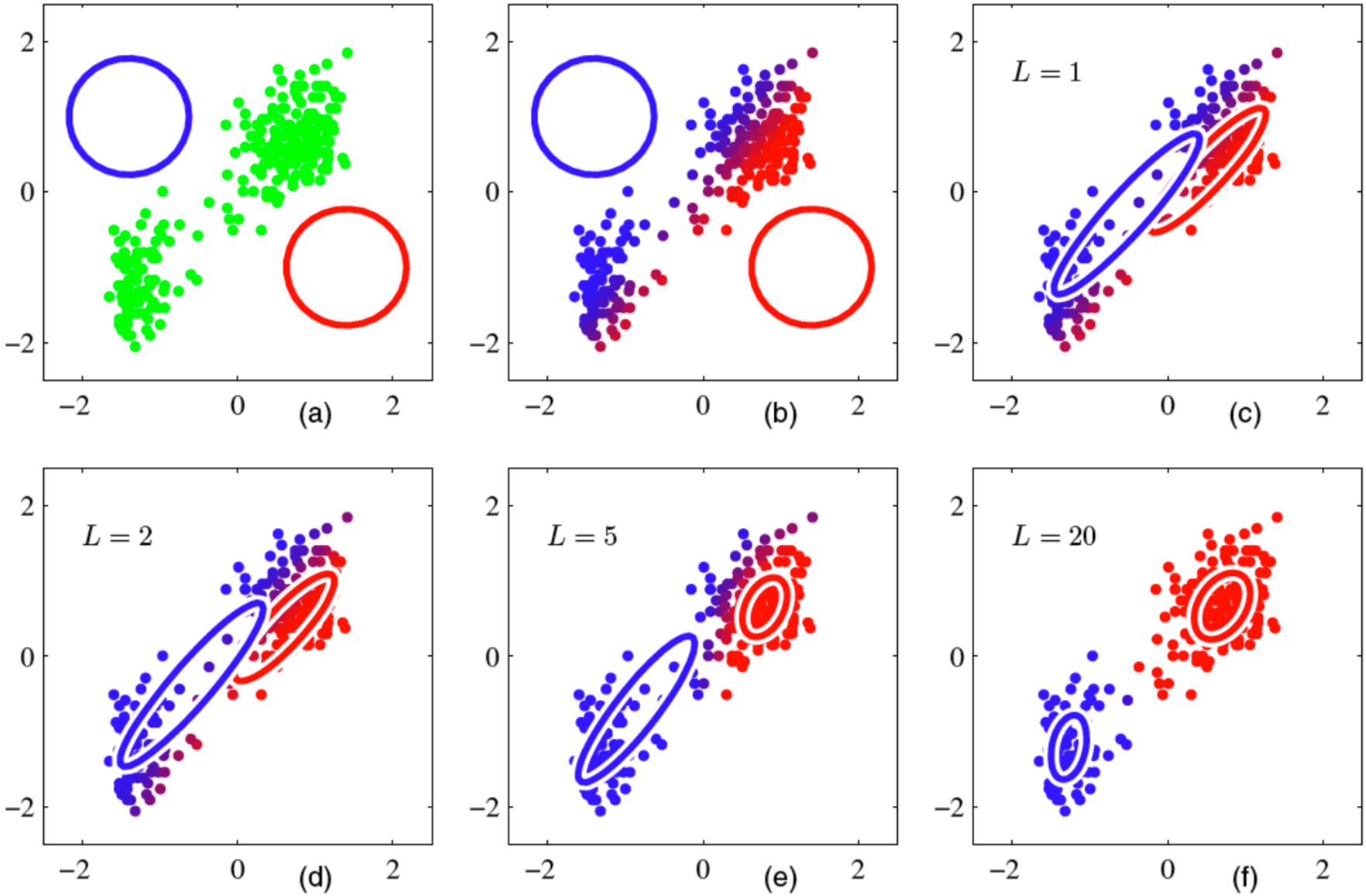
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_k^{(n)} \left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right)\left(\mathbf{x}^{(n)} - \boldsymbol{\mu}_k\right)^\top,$$

$$\pi_k = \frac{N_k}{N}, \text{ with } N_k = \sum_{n=1}^{N} \gamma_k^{(n)}.$$

- Evaluate log likelihood and check for convergence

$$\ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \Big(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^{(n)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Big).$$

EM for fitting GMM



Relation to k-Means

The K-Means Algorithm:

- **Assignment step**: Assign each data point to the closest cluster, *i.e.*, hard assignment
- **Refitting step**: Move each cluster center to the center of gravity of the data assigned to it

The EM Algorithm:

- **E-step**: Given the current model, compute the posterior probability over z for each data point, like soft assignment
- **M-step**: Given the posterior probability, update the model parameters by maximizing the expected log-likelihood

Relation to k-Means

- If fixing the covariance matrices $\mathbf{\Sigma}$ as the identity matrix $\mathbf{I}$ for all Gaussian components, EM for GMMs is reduced to a soft version of K-means.
- Instead of hard assignments in the E-step, EM does **soft assignments** based on the softmax of the squared Euclidean distance from each point to each cluster.
- Each center moved by **weighted means** of the data, with weights given by soft assignments.
- In K-means, weights are 0 or 1.
- GMM provides a probabilistic view of clustering - Each cluster corresponds to a different Gaussian component.

Where does EM come from?

The final issue:

- Let's recall the original objective function (*i.e.*, log-likelihood) of fitting GMM, as follows

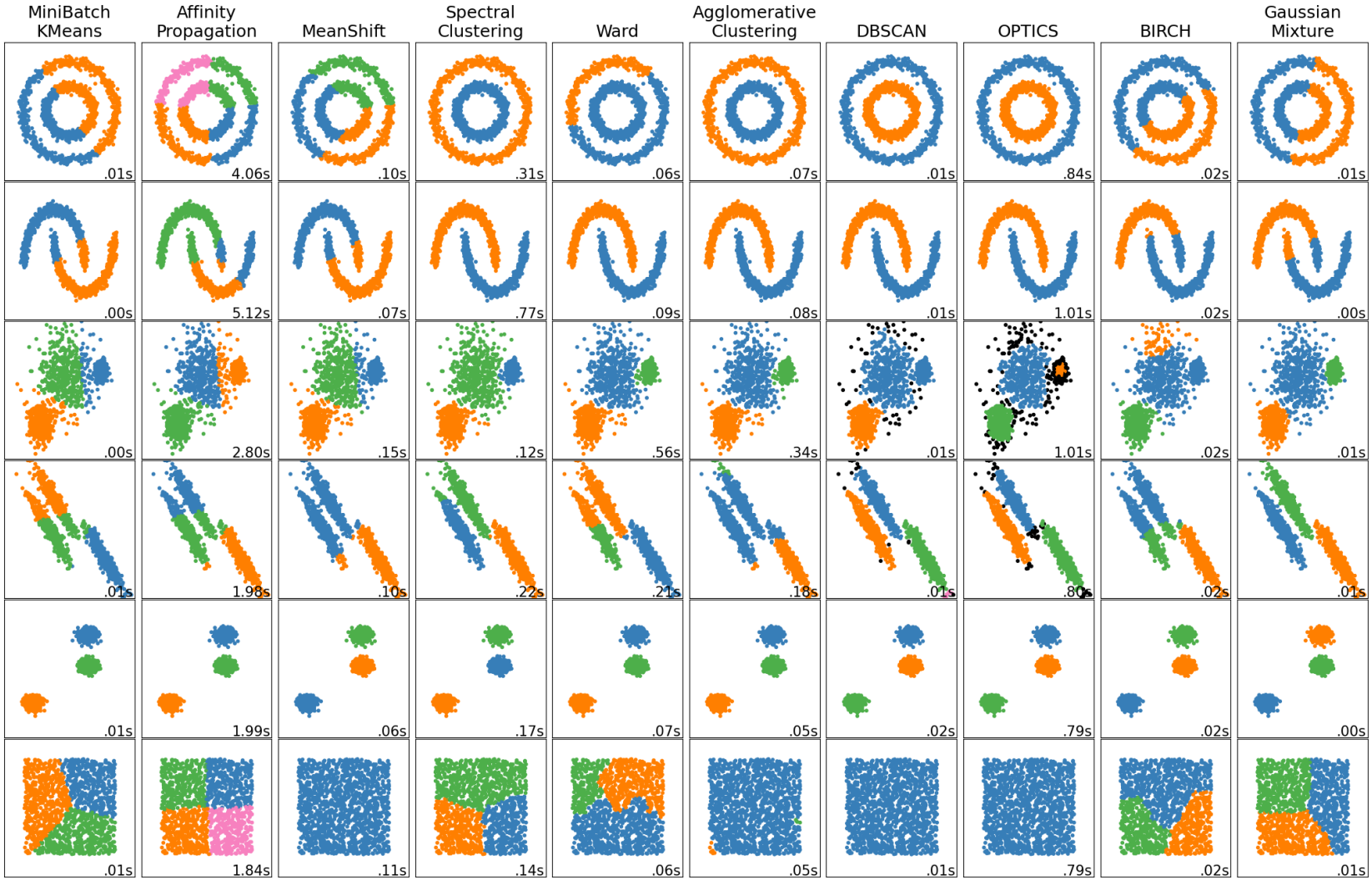
$$\ell(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \sum_{z^{(n)}=1}^{K} p\left(\mathbf{x}^{(n)}, z^{(n)} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right).$$

- However, in the EM algorithm introduced above, this objective function never occurs. Instead, we maximize the following expected log-likelihood in the M-step,

$$\sum_{n=1}^{N} \mathbb{E}_{p(z^{(n)}|\mathbf{x}^{(n)})}\left[\ln\left(p(\mathbf{x}^{(n)}, z^{(n)} \mid \Theta)\right)\right].$$

- So, why EM is a good algorithm for fitting GMM? What is relationship between above two objective functions?
- In the next lecture, we will provide a principled justification of the EM algorithm to answer these questions.

Examples of Clustering Results



Link: https://scikit-learn.org/stable/auto_examples/cluster/plot_clus
comparison.html#sphx-glr-auto-examples-cluster-plot-cluster-compari

Reference

Further readings:

- Chapter 9 in the book "Pattern Recognition and Machine Learning". Link
- Demo with code: https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html#sklearn.mixture.GaussianMix