

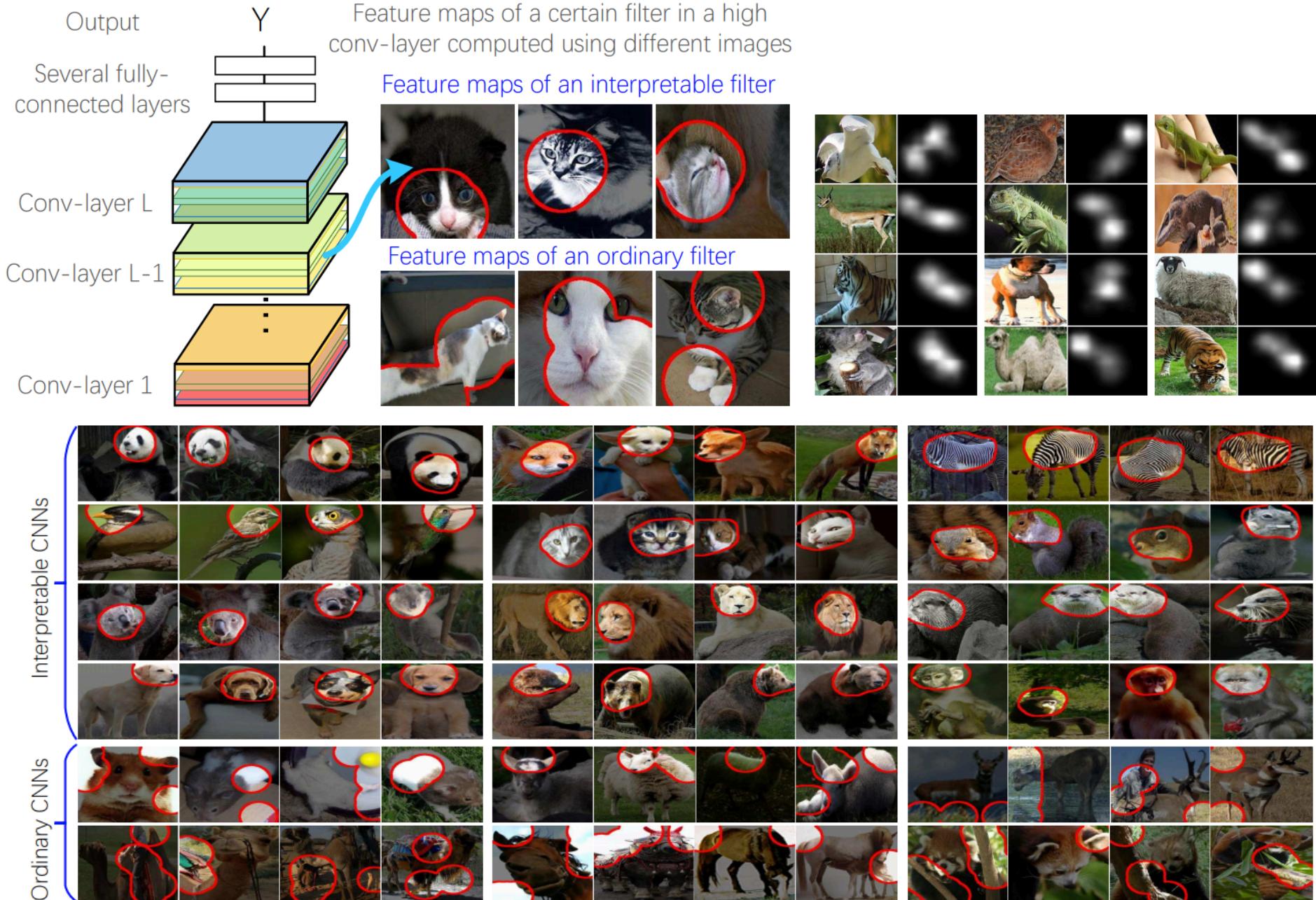
# Density Estimation with Gaussian Mixture Models

Liang Zheng

Australian National University

[liang.zheng@anu.edu.au](mailto:liang.zheng@anu.edu.au)

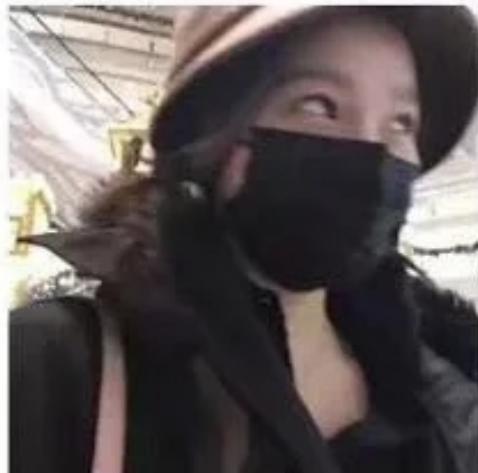
# Interpretable Convolutional Neural Networks Zhang et al., CVPR 2018



# Bayes' theorem

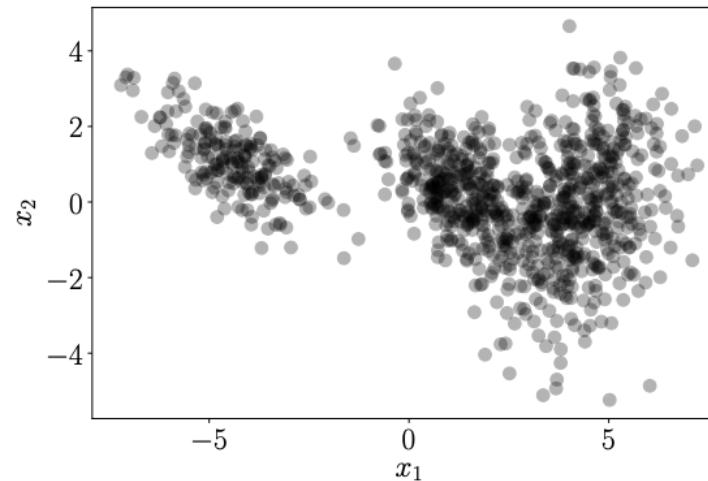


$$p(\theta | y) = \frac{\overbrace{p(y|\theta)}^{\text{posterior}} \overbrace{p(\theta)}^{\text{prior}}}{\overbrace{p(y)}^{\text{evidence}}}$$



# Motivation

- In practice, the Gaussian distribution has limited modeling capabilities.
- Below is a two-dimensional dataset that cannot be meaningfully represented by a single Gaussian

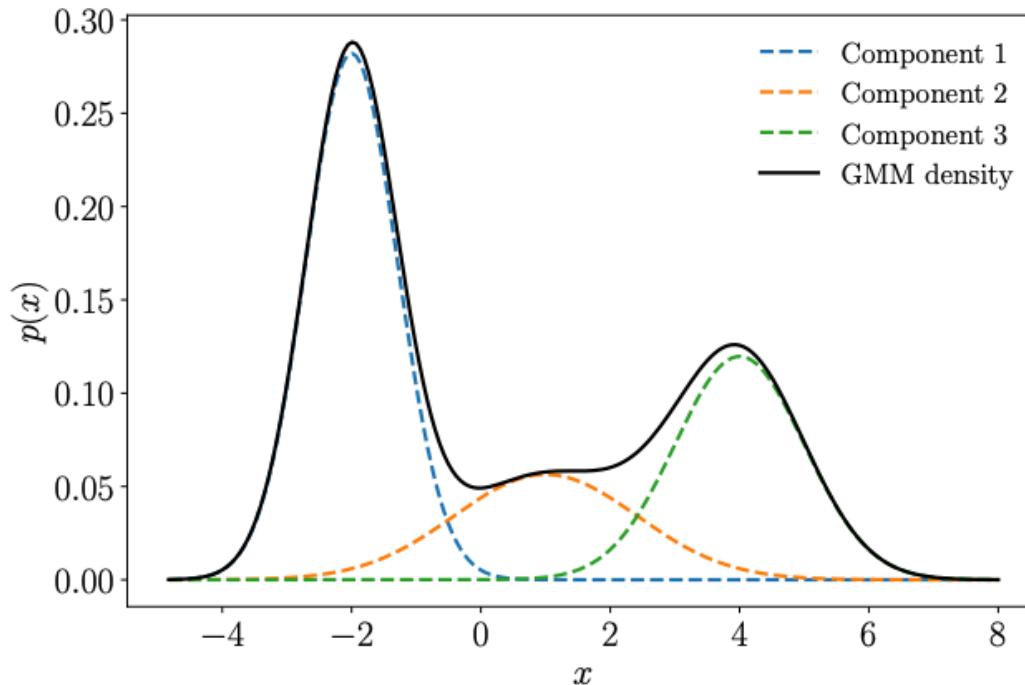


- We can use **mixture models** for density estimation.
- Mixture models can be used to describe a distribution  $p(\mathbf{x})$  by a convex combination of  $K$  simple (base) distributions

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k p_k(\mathbf{x})$$
$$0 \leq \pi_k \leq 1, \quad \sum_{k=1}^K \pi_k = 1$$

where the components  $p_k$  are members of a family of basic distributions, e.g., Gaussians, Bernoullis, or Gammas, and the  $\pi_k$  are mixture weights.

# 11.1 Gaussian Mixture Model



The Gaussian mixture distribution (black) is composed of a convex combination of Gaussian distributions and is more expressive than any individual component. Dashed lines represent the weighted Gaussian components.

$$p(x|\theta) = 0.5\mathcal{N}\left(x \left| -2, \frac{1}{2} \right.\right) + 0.2\mathcal{N}(x|1, 2) + 0.3\mathcal{N}(x|4, 1)$$

# 11.1 Gaussian Mixture Model

- A **Gaussian mixture model (GMM)** is a density model where we combine a finite number of  $K$  Gaussian distributions  $N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$  so that

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^K \pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$
$$0 \leq \pi_k \leq 1, \sum_{k=1}^K \pi_k = 1$$

where we defined  $\boldsymbol{\theta} := \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k: k = 1, \dots, K\}$  as the collection of all parameters of the GMM.

- GMM gives us significantly more flexibility for modeling complex densities than a simple Gaussian distribution.
- **Parameter Learning via Maximum Likelihood**
- Assume we are given a dataset  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , where  $\mathbf{x}_n, n = 1, \dots, N$ , are drawn i.i.d. from an unknown distribution  $p(\mathbf{x})$ . Our objective is to find a good approximation/representation of this unknown distribution  $p(\mathbf{x})$  by means of a GMM with  $K$  components.

## 11.2 Parameter Learning via Maximum Likelihood

- Assume we are given a dataset  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , where  $\mathbf{x}_n, n = 1, \dots, N$ , are drawn i.i.d. from an unknown distribution  $p(\mathbf{x})$ .
- Our objective is to find a good approximation/representation of this unknown distribution  $p(\mathbf{x})$  by means of a GMM with  $K$  components.

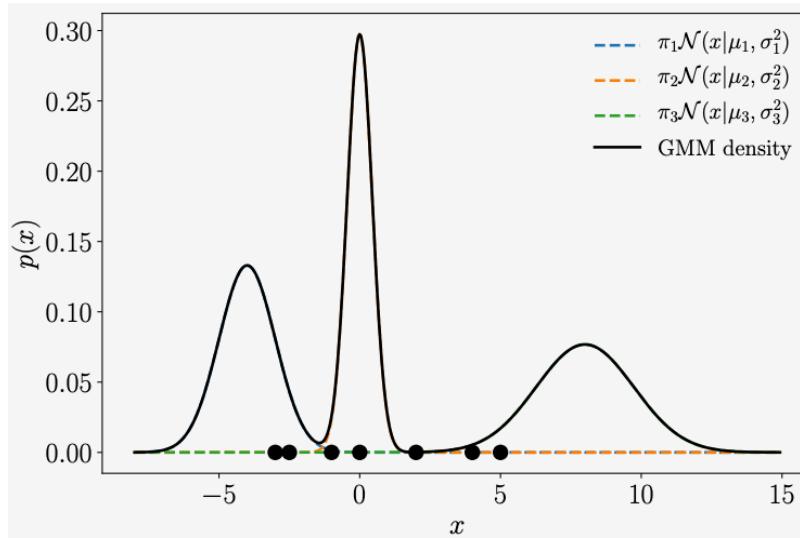
# Example

- We consider a one-dimensional dataset  $\mathcal{X} = \{-3, -2.5, -1, 0, 2, 4, 5\}$  consisting of 7 data points and wish to find a GMM with  $K = 3$  components that models the density of the data.
- We initialize the mixture components as

$$\begin{aligned} p_1(x) &= \mathcal{N}(x| -4, 1) \\ p_2(x) &= \mathcal{N}(x| 0, 0.2) \\ p_3(x) &= \mathcal{N}(x| 8, 3) \end{aligned}$$

and assign them equal weights  $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$ .

- We can view the corresponding model and the data points below.



- How to obtain a maximum likelihood estimate  $\boldsymbol{\theta}_{ML}$  of model parameters  $\boldsymbol{\theta}$ ?
- We start by writing down the likelihood, i.e., the predictive distribution of the training data given the parameters. We exploit our i.i.d. assumption, which leads to the factorized likelihood

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{n=1}^N p(x_n|\boldsymbol{\theta}), \quad p(x_n|\boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Observed data      
  Mixture component  
 Mixture proportion

where every individual likelihood term  $p(x_n|\boldsymbol{\theta})$  is a Gaussian mixture density.

- Then we obtain the log-likelihood (loss function) as

$$\mathcal{L}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k) = \log p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{n=1}^N \log p(x_n|\boldsymbol{\theta}) = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- We aim to find parameters  $\boldsymbol{\theta}_{ML}^*$  (including  $\boldsymbol{\mu}_k^*, \boldsymbol{\Sigma}_k^*, \pi_k^*$ ) that maximize log-likelihood  $\mathcal{L}$  defined above.

- We obtain the following necessary conditions when we optimize the log-likelihood with respect to the GMM parameters  $\mu_k, \Sigma_k, \pi_k$ :

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mu_k} = \mathbf{0}^T &\Leftrightarrow \sum_{n=1}^N \frac{\partial \log p(x_n | \theta)}{\partial \mu_k} = \mathbf{0}^T \\ \frac{\partial \mathcal{L}}{\partial \Sigma_k} = 0 &\Leftrightarrow \sum_{n=1}^N \frac{\partial \log p(x_n | \theta)}{\partial \Sigma_k} = \mathbf{0} \\ \frac{\partial \mathcal{L}}{\partial \pi_k} = 0 &\Leftrightarrow \sum_{n=1}^N \frac{\partial \log p(x_n | \theta)}{\partial \pi_k} = 0\end{aligned}$$

- For all three necessary conditions, by applying the chain rule, we require partial derivatives of the form

$$\frac{\partial \log p(x_n | \theta)}{\partial \theta} = \frac{1}{p(x_n | \theta)} \frac{\partial p(x_n | \theta)}{\partial \theta}$$

- where  $\theta = \{\mu_k, \Sigma_k, \pi_k: k = 1, \dots, K\}$  are the model parameters and

$$\frac{1}{p(x_n | \theta)} = \frac{1}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}$$

## 11.2.1 Responsibilities

- We define the quantity

$$r_{nk} := \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}$$

as the responsibility of the  $k$ th mixture component for the  $n$ th data point.

- We can see  $r_{nk}$  is proportional to the likelihood

$$p(x_n | \pi_k, \mu_k, \Sigma_k) = \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)$$

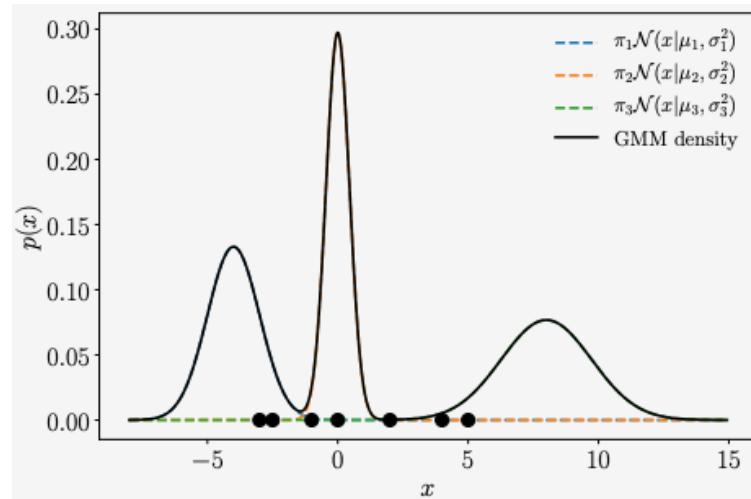
of the  $k$ th mixture component given the data point.

- The responsibility  $r_{nk}$  represents the posterior probability that  $x_n$  has been generated by the  $k$ th mixture component
- Note that  $\mathbf{r}_n := [r_{n1}, \dots, r_{nK}]^\top \in \mathbb{R}^K$  is a (normalized) probability vector, i.e.,  $\sum_k r_{nk} = 1$  with  $r_{nk} \geq 0$ .
- This probability vector distributes probability mass among the  $K$  mixture components, and we can think of  $\mathbf{r}_n$  as a “soft assignment” of  $x_n$  to the  $K$  mixture components.

# Example - Responsibilities

- From the figure below, we compute the responsibilities  $r_{nk}$

$$\begin{matrix} \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.057 & 0.943 & 0.0 \\ 0.001 & 0.999 & 0.0 \\ 0.0 & 0.066 & 0.934 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \in \mathbb{R}^{N \times K} \end{matrix}$$



- The  $n$ th row tells us the responsibilities of all mixture components for  $x_n$ .
- The sum of all  $K$  responsibilities for a data point (sum of every row) is 1.
- The  $k$ th column gives us an overview of the responsibility of the  $k$ th mixture component.
- The third mixture component (third column) is not responsible for any of the first four data points, but takes much responsibility of the remaining data points.
- The sum of all entries of a column gives us the values  $N_k$ , i.e., the total responsibility of the  $k$ th mixture component. In our example, we get  $N_1 = 2.058$ ,  $N_2 = 2.008$ ,  $N_3 = 2.934$ .
- We will determine the updates of the model parameters  $\mu_k$ ,  $\Sigma_k$ , and  $\pi_k$  for given responsibilities

## 11.3 EM Algorithm

- In GMM, we first initialize the parameters  $\mu_k$ ,  $\Sigma_k$ , and  $\pi_k$  and alternate until convergence between the following two steps
- E-step: Evaluate the responsibilities  $r_{nk}$  (probability of data point  $n$  belonging to mixture component  $k$ )
- M-step: Use the updated responsibilities to re-estimate the parameters  $\mu_k$ ,  $\Sigma_k$ , and  $\pi_k$

# Check your understanding

- Given a dataset generated by a mixture of 3 Gaussians, when we randomly sample a data point, it has the probability of 1/3 belonging to each Gaussian.
- A GMM is a linear combination of several Gaussian distributions.
- In GMM, K (number of Gaussians) is a hyperparameter.
- If a dataset is not generated by Gaussian distributions, it cannot be modeled by GMM.

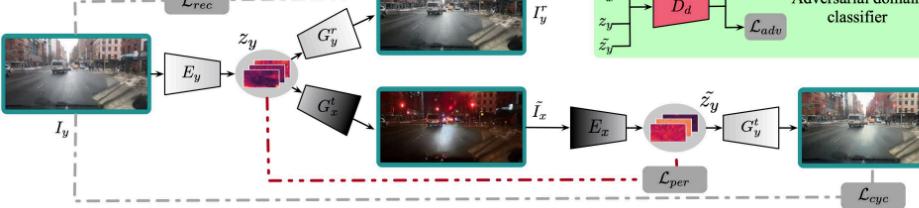
# ForkGAN: Seeing into the Rainy Night. Zheng et al., ECCV 2020



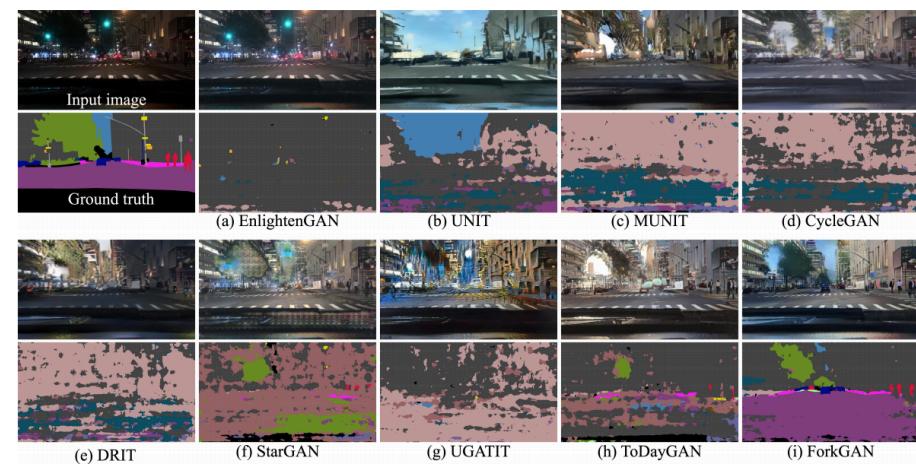
Night to day



Day to Night



(a) Training



(a) Training

(b) (c) (d) (e) (f) (g) (h) (i)

# 11.3 EM Algorithm

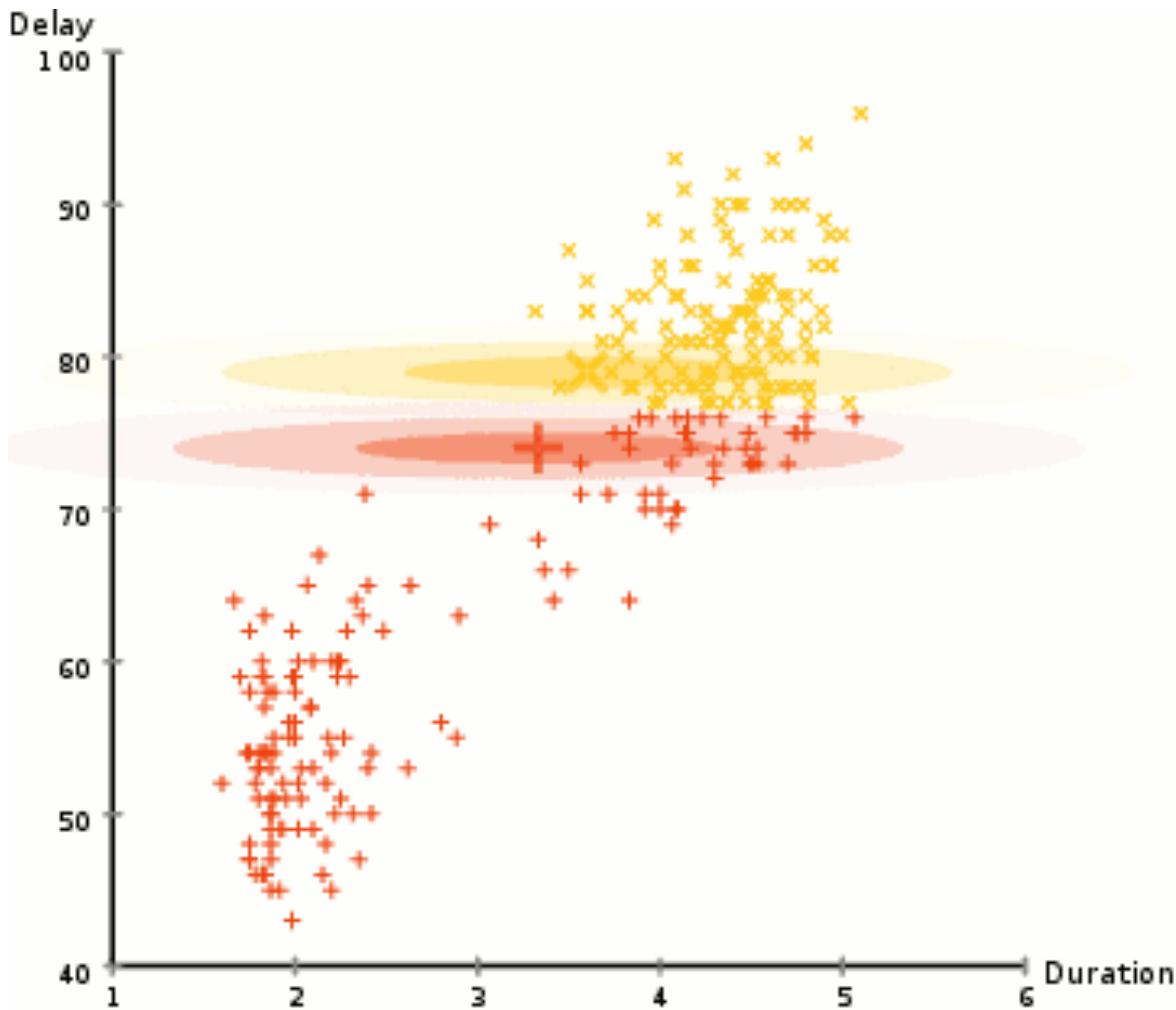
- Initialize  $\mu_k, \Sigma_k, \pi_k$ . (below is an example)
  - $\pi_k = 1/K$  for all  $k$
  - $\mu_k$ : centroids from  $k$ -means algorithm or using randomly chosen data points
  - $\Sigma_k$  =  $\Sigma$  the sample variance, for all  $k$
- E-step: Evaluate responsibilities  $r_{nk}$  for every data point  $x_n$  using current parameters  $\pi_k, \mu_k, \Sigma_k$ :
$$r_{nk} = \frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}$$
- M-step: Re-estimate parameters  $\pi_k, \mu_k, \Sigma_k$  using the current responsibilities  $r_{nk}$  (from E-step):

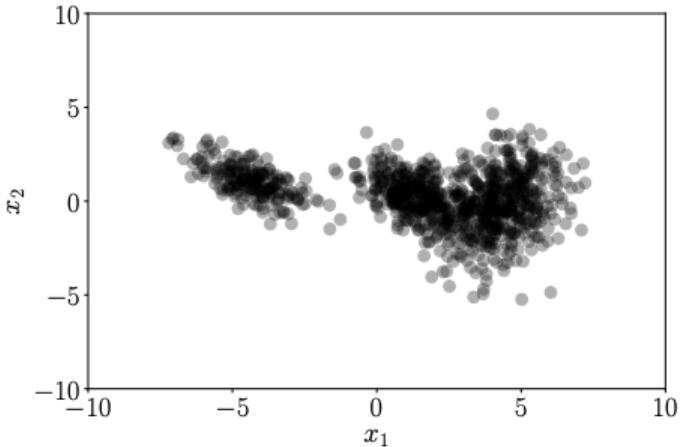
$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} x_n$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

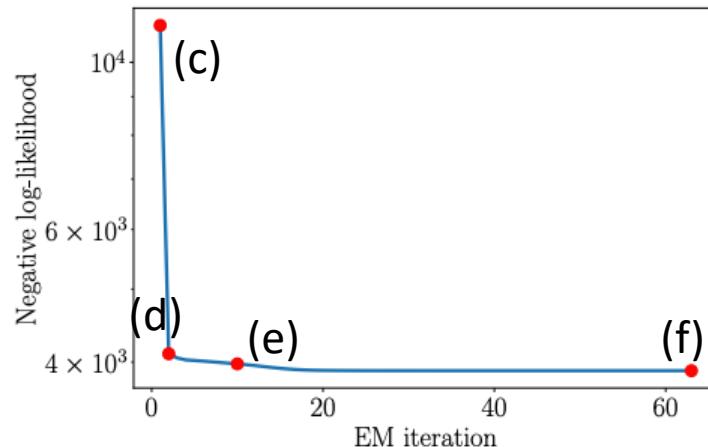
$$\pi_k = \frac{N_k}{N}$$

## 11.3 EM Algorithm

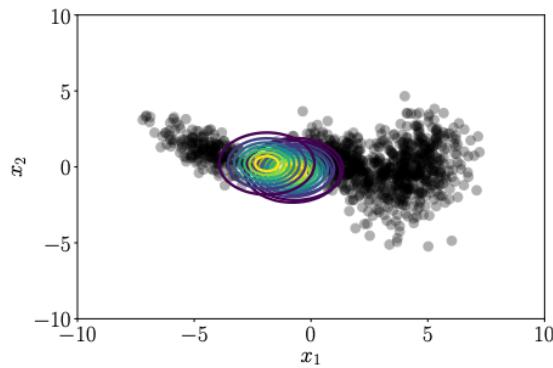




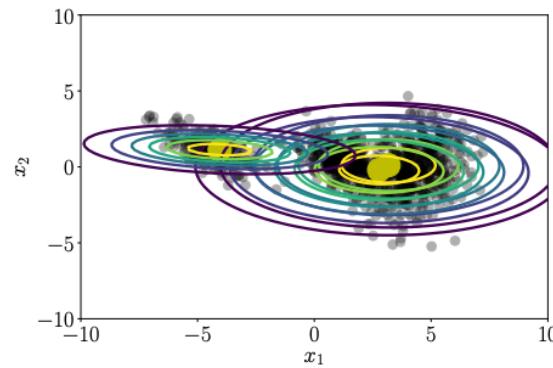
(a) Dataset.



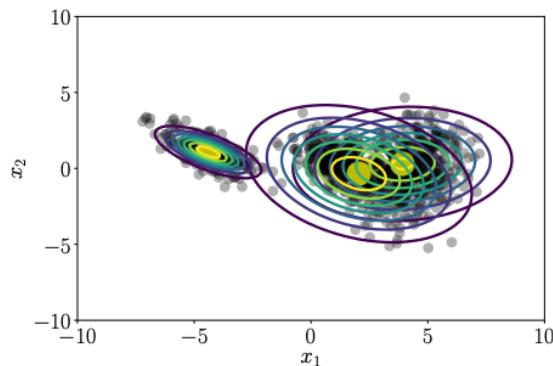
(b) Negative log-likelihood.



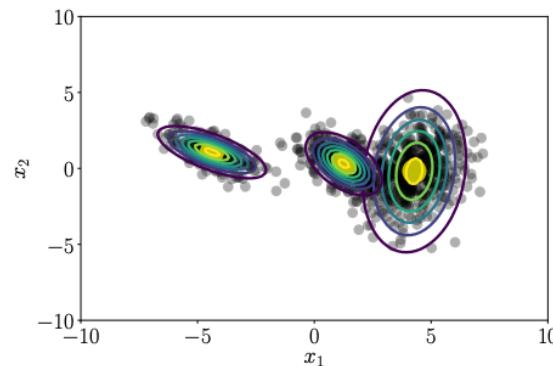
(c) EM initialization.



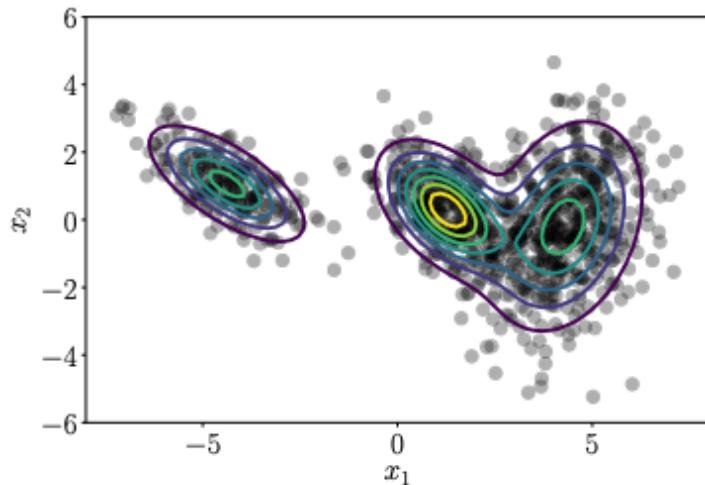
(d) EM after one iteration.



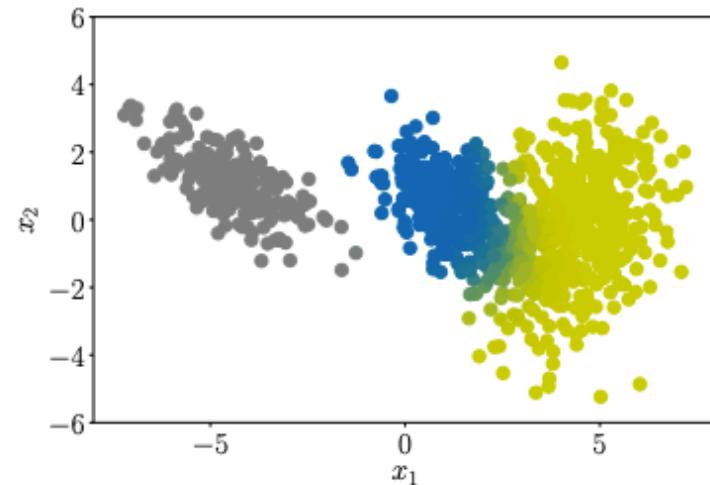
(e) EM after 10 iterations.



(f) EM after 62 iterations.



(a) GMM fit after 62 iterations.



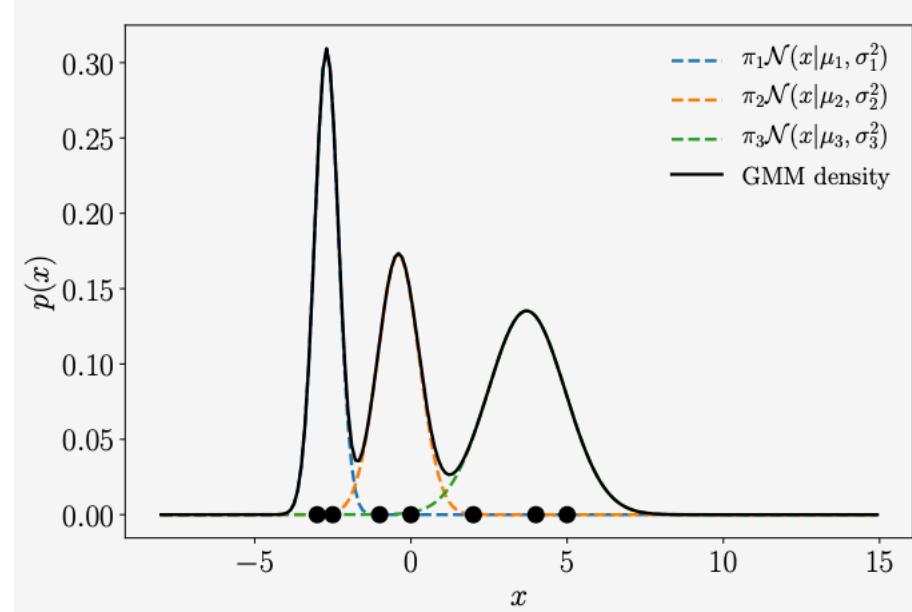
(b) Dataset colored according to the responsibilities of the mixture components.

- The dataset is colored according to the responsibilities of the mixture components when EM converges.
- A single mixture component is highly responsible for the data on the left.
- The overlap of the two data clusters on the right could have been generated by two mixture components.
- It becomes clear that there are data points that cannot be uniquely assigned to a single component (either **blue** or **yellow**), such that the responsibilities of these two clusters for those points are around **0.5**.

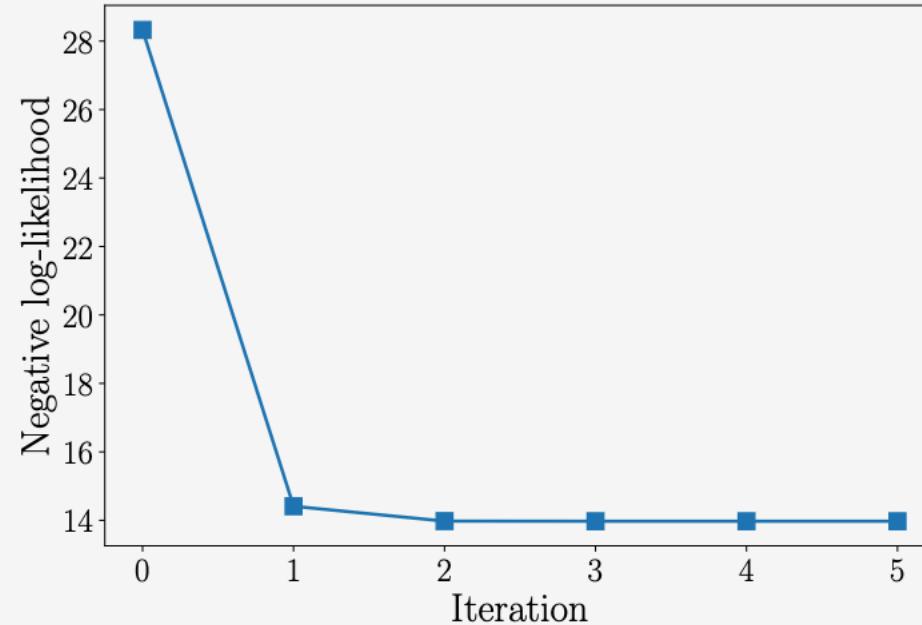
# 11.3 EM Algorithm

- The final GMM is given as

$$p(x) = 0.29\mathcal{N}(x|-2.75, 0.06) + 0.28\mathcal{N}(x|-0.50, 0.25) + 0.43\mathcal{N}(x|3.64, 1.63)$$



Final GMM fit. After five iterations, the EM algorithm converges and returns this GMM



Negative log-likelihood as a function of the EM iterations.

## 11.2.2 Updating the Means

- The update of the mean parameters  $\mu_k, k = 1, \dots, K$ , of the GMM is given by

$$\mu_k^{new} = \frac{\sum_{n=1}^N r_{nk} x_n}{\sum_{n=1}^N r_{nk}}$$

- Proof: Calculate the gradient of the log-likelihood with respect to  $\mu_k$
- Considering

$$\begin{aligned}\mathcal{L}(\mu_k, \Sigma_k, \pi_k) &= \log p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{n=1}^N \log p(x_n|\boldsymbol{\theta}) \\ p(x_n|\boldsymbol{\theta}) &= \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\end{aligned}$$

- We have

$$\frac{\partial p(x_n|\boldsymbol{\theta})}{\partial \mu_k} = \sum_{j=1}^K \pi_j \frac{\partial \mathcal{N}(x_n|\mu_j, \Sigma_j)}{\partial \mu_k} = \pi_k \frac{\partial \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\partial \mu_k}$$

- Recall our knowledge in multivariate Gaussian distribution and vector calculus

$$\begin{aligned}p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})) \\ \frac{\partial \mathbf{x}^T \mathbf{B} \mathbf{x}}{\partial \mathbf{x}} &= \mathbf{x}^T (\mathbf{B} + \mathbf{B}^T)\end{aligned}$$

- We have

$$\frac{\partial p(x_n|\boldsymbol{\theta})}{\partial \mu_k} = \pi_k (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} \mathcal{N}(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

## 11.2.2 Updating the Means

- The desired partial derivative of  $\mathcal{L}$  with respect to  $\mu_k$  is given as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mu_k} &= \sum_{n=1}^N \frac{\partial \log p(x_n | \theta)}{\partial \mu_k} = \sum_{n=1}^N \frac{1}{p(x_n | \theta)} \frac{\partial p(x_n | \theta)}{\partial \mu_k}, \\ &= \sum_{n=1}^N (x_n - \mu_k)^T \Sigma_k^{-1} \underbrace{\frac{\pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)}}_{= r_{nk}} = r_{nk} \\ &= \sum_{n=1}^N r_{nk} (x_n - \mu_k)^T \Sigma_k^{-1} \end{aligned}$$

- We now solve the above gradient for  $\mu_k^{new}$  so that  $\frac{\partial \mathcal{L}(\mu_k^{new})}{\partial \mu_k} = \mathbf{0}^T$  and obtain

$$\sum_{n=1}^N r_{nk} x_n = \sum_{n=1}^N r_{nk} \mu_k^{new} \Leftrightarrow \mu_k^{new} = \frac{\sum_{n=1}^N r_{nk} x_n}{\sum_{n=1}^N r_{nk}} = \frac{1}{N_k} \sum_{n=1}^N r_{nk} x_n$$

where we define

$$N_k := \sum_{n=1}^N r_{nk}$$

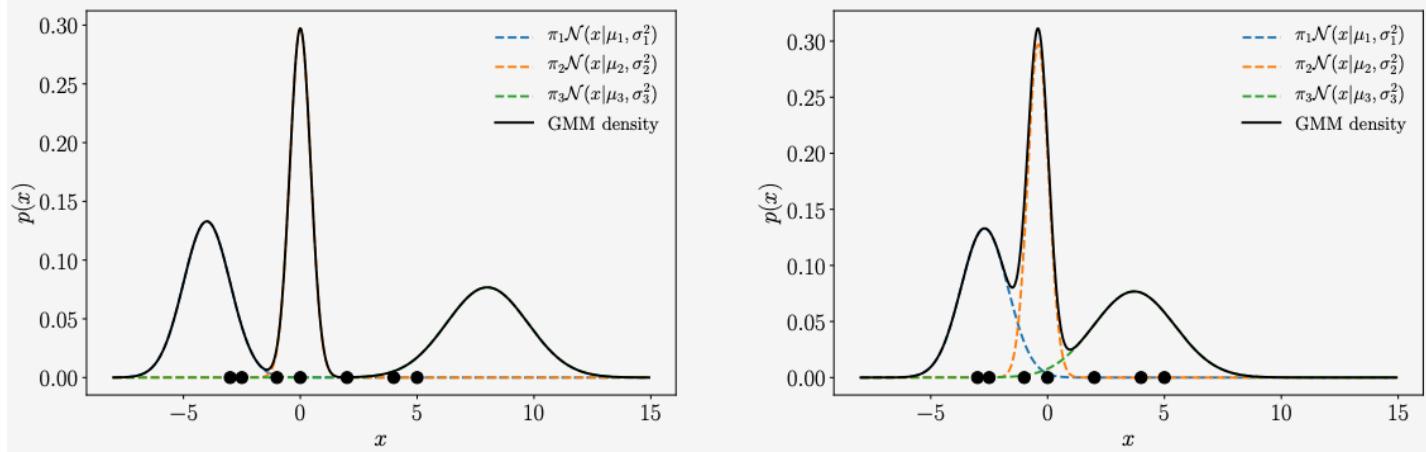
as the total responsibility of the  $k$ th mixture component for the entire dataset.

- This concludes the proof.

## 11.2.2 Updating the Means

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_{n=1}^N r_{nk} \mathbf{x}_n}{\sum_{n=1}^N r_{nk}}$$

- This is an importance-weighted Monte Carlo estimate of the mean.
- The importance weights of data point  $\mathbf{x}_n$  is  $r_{nk}$
- Mean update



Initialization:

$$\mathcal{X} = \{-3, -2.5, -1, 0, 2, 4, 5\}$$

$$\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$$

$$p_1(x) = \mathcal{N}(x|-4, 1)$$

$$p_2(x) = \mathcal{N}(x|0, 0.2)$$

$$p_3(x) = \mathcal{N}(x|8, 3)$$

$$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.057 & 0.943 & 0.0 \\ 0.001 & 0.999 & 0.0 \\ 0.0 & 0.066 & 0.934 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$\mu_1 : -4 \rightarrow -2.7$$

$$\mu_2 : 0 \rightarrow -0.4$$

$$\mu_3 : 8 \rightarrow 3.7$$

$$-2.7 = \frac{-3 \times 1 - 2.5 \times 1 - 1 \times 0.057 - 0 \times 0.001}{1 + 1 + 0.057 + 0.001}$$

## 11.2.3 Updating the Covariances

- The update of the covariance parameters  $\Sigma_k, k = 1, \dots, K$  is given by

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

- Proof* We compute the partial derivatives of the log-likelihood  $\mathcal{L}$  with respect to the covariances  $\Sigma_k$ , set them to  $\mathbf{0}$ , and solve for  $\Sigma_k$ . We start by

$$\frac{\partial \mathcal{L}}{\partial \Sigma_k} = \sum_{n=1}^N \frac{\partial \log p(\mathbf{x}_n | \boldsymbol{\theta})}{\partial \Sigma_k} = \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n | \boldsymbol{\theta})} \frac{\partial p(\mathbf{x}_n | \boldsymbol{\theta})}{\partial \Sigma_k}$$

- We already know  $1/p(\mathbf{x}_n | \boldsymbol{\theta})$ . To obtain  $\partial p(\mathbf{x}_n | \boldsymbol{\theta})/\partial \Sigma_k$ , we have,

$$\begin{aligned} \frac{\partial p(\mathbf{x}_n | \boldsymbol{\theta})}{\partial \Sigma_k} &= \frac{\partial}{\partial \Sigma_k} \left( \pi_k (2\pi)^{-\frac{D}{2}} \det(\Sigma_k)^{-\frac{1}{2}} \exp \left( -\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \right) \right) \\ &= \pi_k (2\pi)^{-\frac{D}{2}} \left[ \frac{\partial}{\partial \Sigma_k} \det(\Sigma_k)^{-\frac{1}{2}} \exp \left( -\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \right) \right. \\ &\quad \left. + \det(\Sigma_k)^{-\frac{1}{2}} \frac{\partial}{\partial \Sigma_k} \exp \left( -\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \right) \right] \end{aligned}$$

## 11.2.3 Updating the Covariances

- From Vector Calculus, we have the following identities

$$\frac{\partial}{\partial \Sigma_k} \det(\Sigma_k)^{-\frac{1}{2}} = -\frac{1}{2} \det(\Sigma_k)^{-\frac{1}{2}} \Sigma_k^{-1}$$

$$\frac{\partial}{\partial \Sigma_k} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) = -\Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}$$

- We obtain the desired partial derivative

$$\frac{\partial p(\mathbf{x}_n | \boldsymbol{\theta})}{\partial \Sigma_k} = \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k) \cdot \left[ -\frac{1}{2} (\Sigma_k^{-1} - \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}) \right]$$

- Thus, the partial derivative of the log-likelihood with respect to  $\Sigma_k$  is given by

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \Sigma_k} &= \sum_{n=1}^N \frac{\partial \log p(\mathbf{x}_n | \boldsymbol{\theta})}{\partial \Sigma_k} = \sum_{n=1}^N \frac{1}{p(\mathbf{x}_n | \boldsymbol{\theta})} \frac{\partial p(\mathbf{x}_n | \boldsymbol{\theta})}{\partial \Sigma_k} \\ &= \underbrace{\sum_{n=1}^N \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \Sigma_j)} \cdot \left[ -\frac{1}{2} (\Sigma_k^{-1} - \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}) \right]}_{= r_{nk}} \\ &= -\frac{1}{2} \sum_{n=1}^N r_{nk} (\Sigma_k^{-1} - \Sigma_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}) \\ &= -\frac{1}{2} \Sigma_k^{-1} \underbrace{\sum_{n=1}^N r_{nk}}_{N_k} + \frac{1}{2} \Sigma_k^{-1} \left( \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \right) \Sigma_k^{-1} \end{aligned}$$

## 11.2.3 Updating the Covariances

- Setting this partial derivative to  $\mathbf{0}$ , we obtain the necessary optimality condition

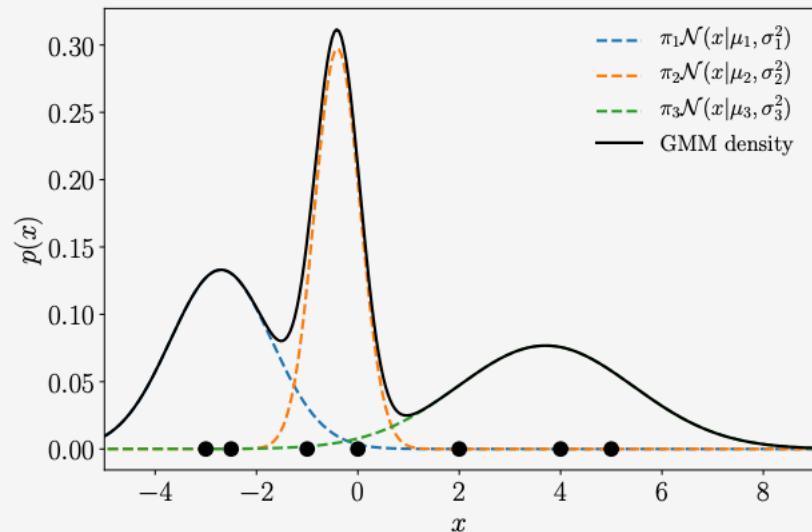
$$N_k \Sigma_k^{-1} = \Sigma_k^{-1} \left( \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \right) \Sigma_k^{-1}$$
$$\Leftrightarrow N_k \mathbf{I} = \left( \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \right) \Sigma_k^{-1}$$

- By solving for  $\Sigma_k$ , we obtain

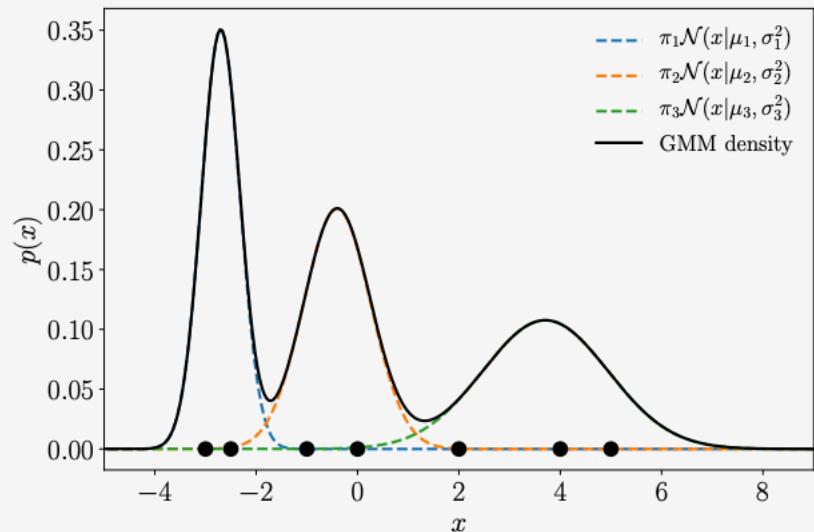
$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

- This gives us a simple update rule for  $\Sigma_k$  for  $k = 1, \dots, K$  and proves our theorem.
- This update method is the **weighted** covariance of data points  $\mathbf{x}_n$  associated with the  $k$ th component.
- The **weights** are the responsibilities  $r_{nk}$

## 11.2.3 Updating the Covariances



(a) GMM density and individual components prior to updating the variances.



(b) GMM density and individual components after updating the variances.

$$\begin{aligned}\sigma_1^2 &: 1 \rightarrow 0.14 \\ \sigma_2^2 &: 0.2 \rightarrow 0.44 \\ \sigma_3^2 &: 3 \rightarrow 1.53\end{aligned}$$

## 11.2.4 Updating the Mixture Weights

- The mixture weights of the GMM are updated as

$$\pi_k^{new} = \frac{N_k}{N}, k = 1, \dots, K$$

where  $N$  is the number of data points

- Proof* We calculate the partial derivative of the log-likelihood with respect to the weight parameters  $\pi_k, k = 1, \dots, K$ .
- We have the constraint

$$\sum_k \pi_k = 1$$

- Using Lagrange multipliers (will not be covered in this course), we have

$$\begin{aligned}\mathfrak{L} &= \mathcal{L} + \lambda \left( \sum_{k=1}^K \pi_k - 1 \right) \\ &= \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) + \lambda \left( \sum_{k=1}^K \pi_k - 1 \right)\end{aligned}$$

$$\mathfrak{L} = \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(x_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \lambda \left( \sum_{k=1}^K \pi_k - 1 \right)$$

- We obtain the partial derivative with respect to  $\pi_k$  as

$$\begin{aligned} \frac{\partial \mathfrak{L}}{\partial \pi_k} &= \sum_{n=1}^N \frac{\mathcal{N}(x_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda \\ &= \underbrace{\frac{1}{\pi_k} \sum_{n=1}^N \frac{\pi_k \mathcal{N}(x_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(x_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{= N_k} + \lambda = \frac{N_k}{\pi_k} + \lambda \end{aligned}$$

- The partial derivative with respect to the Lagrange multiplier  $\lambda$  is

$$\frac{\partial \mathfrak{L}}{\partial \lambda} = \sum_{k=1}^K \pi_k - 1$$

- Setting both partial derivatives to 0 yields the system of equations

$$\begin{aligned} \pi_k &= -\frac{N_k}{\lambda} \\ 1 &= \sum_{k=1}^K \pi_k \end{aligned}$$

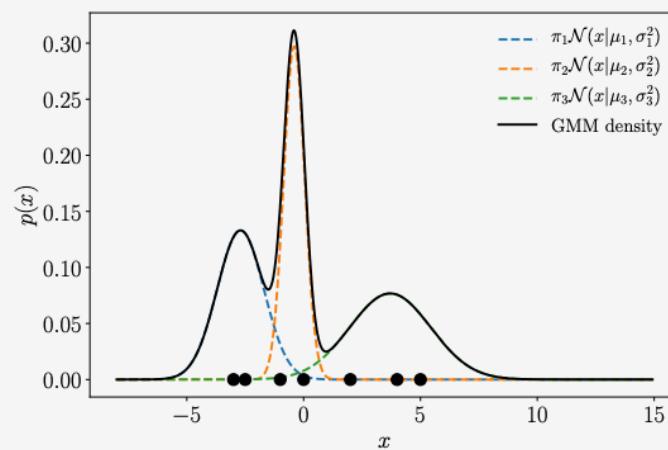
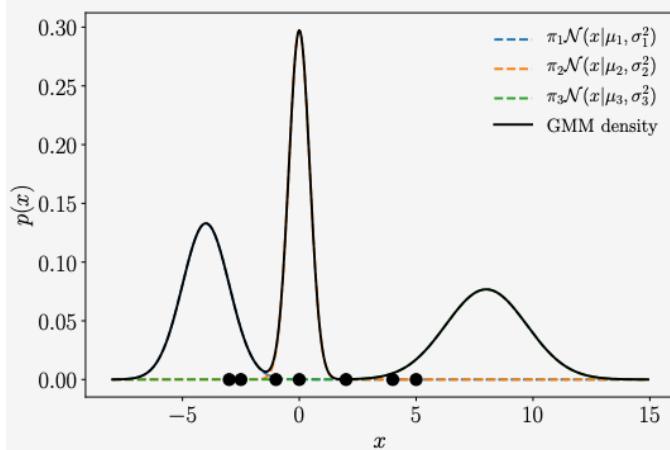
- Using the two equations, we obtain

$$\sum_{k=1}^K \pi_k = 1 \Leftrightarrow - \sum_{k=1}^K \frac{N_k}{\lambda} = 1 \Leftrightarrow - \frac{N}{\lambda} = 1 \Leftrightarrow \lambda = -N$$

- This allows us to substitute  $-N$  for  $\lambda$  in  $\pi_k = -\frac{N_k}{\lambda}$  to obtain

$$\pi_k^{new} = \frac{N_k}{N}$$

which gives us the update for the weight parameters  $\pi_k$  and proves the Theorem.



$$\begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \\ 0.057 & 0.943 & 0.0 \\ 0.001 & 0.999 & 0.0 \\ 0.0 & 0.066 & 0.934 \\ 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$\begin{aligned} \pi_1 & : \frac{1}{3} \rightarrow 0.29(11.50) \\ \pi_2 & : \frac{1}{3} \rightarrow 0.29(11.51) \\ \pi_3 & : \frac{1}{3} \rightarrow 0.42(11.52) \end{aligned}$$

$$0.29 = \frac{1 + 1 + 0.057 + 0.001}{7}$$

- We see that the third component gets more weight/importance, while the other components become slightly less important.

# Generating a new dataset with GMM

- For a given GMM with parameters  $\mu_k, \Sigma_k, \pi_k, k = 1, \dots, K$ , we want to generate a dataset with  $N$  data points.
- We sample an index  $k$  from  $\{1, 2, \dots, K\}$  with probabilities  $\pi_1, \dots, \pi_K$
- We generate a number of  $N\pi_k$  data points for the  $k$ th component
- In the  $k$ th component, every data point is sampled as  $x \sim \mathcal{N}(\mu_k, \Sigma_k)$

# Comparising GMM with K-Means

## Algorithms.

1. k-Means
    - a. Given hard labels, compute centroids
    - b. Given centroids, compute hard labels
  2. GMM
    - a. Given soft labels, compute Gaussians
    - b. Given Gaussians, compute soft labels
- 
- Like k-means, GMM may get stuck in local minima.
  - Unlike k-means, the local minima are more favorable because soft labels allow points to move between clusters slowly.

# Check your understanding

- If  $K$  takes a greater value, the likelihood becomes greater after convergence.
- Assume we have  $N$  data points. The maximum likelihood will be achieved if we set  $K = N$ .
- In GMM, the EM algorithm gives us global minimum, because we can update  $\pi_k$ ,  $\mu_k$  and  $\Sigma_k$  through closed-form solutions.
- GMM has a higher computational complexity than kmeans.
- When the  $N$  data points are close to each other in the feature space, we should set  $K$  to a small value.