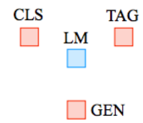
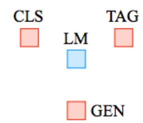
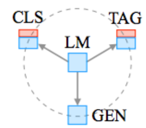
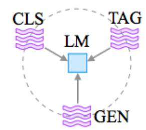
Density Estimation with Gaussian Mixture Models

Liang Zheng
Australian National University
liang.zheng@anu.edu.au

Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. Liu et al., ACM Computing Surveys, 2022.

Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Features (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

Fine-tuning:
Given a pre-trained model,
Classify the sentiment of each sentence into *positive, negative, neutral*

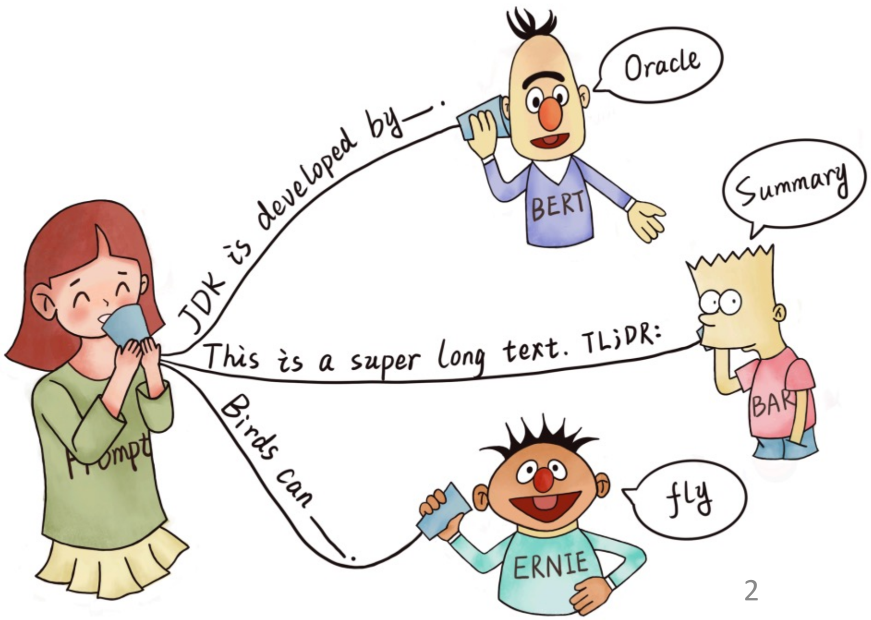
"I love this movie."
label: *positive*

Prompt and predict:
Given a pre-trained model,
fill in the blank:
"I love this movie;
Overall, it was a __ movie."

Your algorithm will fill "excellent", "great", etc.

Then, predict:
"excellent", "great" -> *positive*

Pre-training process:



What we learned last time

$$\underbrace{p(\boldsymbol{x} \mid \boldsymbol{y})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{y} \mid \boldsymbol{x})}^{\text{likelihood}} \; \overbrace{p(\boldsymbol{x})}^{\text{prior}}}{\underbrace{p(\boldsymbol{y})}_{\text{evidence}}}$$

Suppose you are a detective

You observe a crime scene characterized by evidence/clues $\boldsymbol{y}$

Given the scene, you want to know who the criminal is: $p(\boldsymbol{x} \mid \boldsymbol{y})$

You have 2 suspects $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$:

You ask yourself: how likely will $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$ generate the crime scene? $p(\boldsymbol{y} \mid \boldsymbol{x})$

Does the fingerprint in $\boldsymbol{y}$ match $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$?

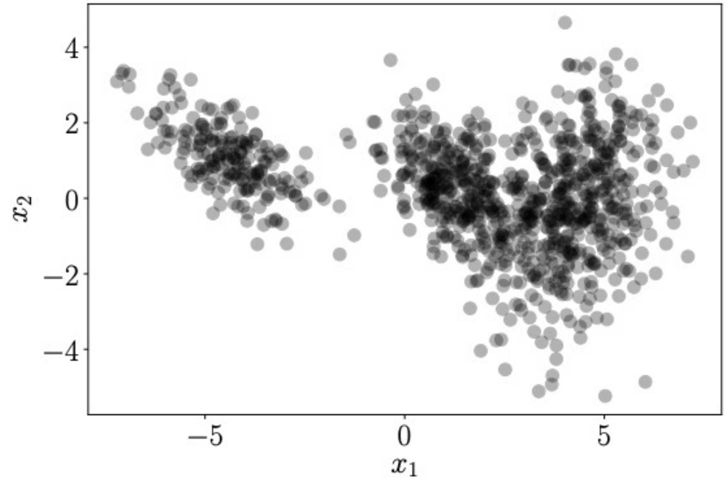
Does the surveillance camera imaging match $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$?

You then ask yourself: how likely will $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$ commit a crime? $p(\boldsymbol{x})$

Do $\boldsymbol{x}_1$ or $\boldsymbol{x}_2$ have a history?

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



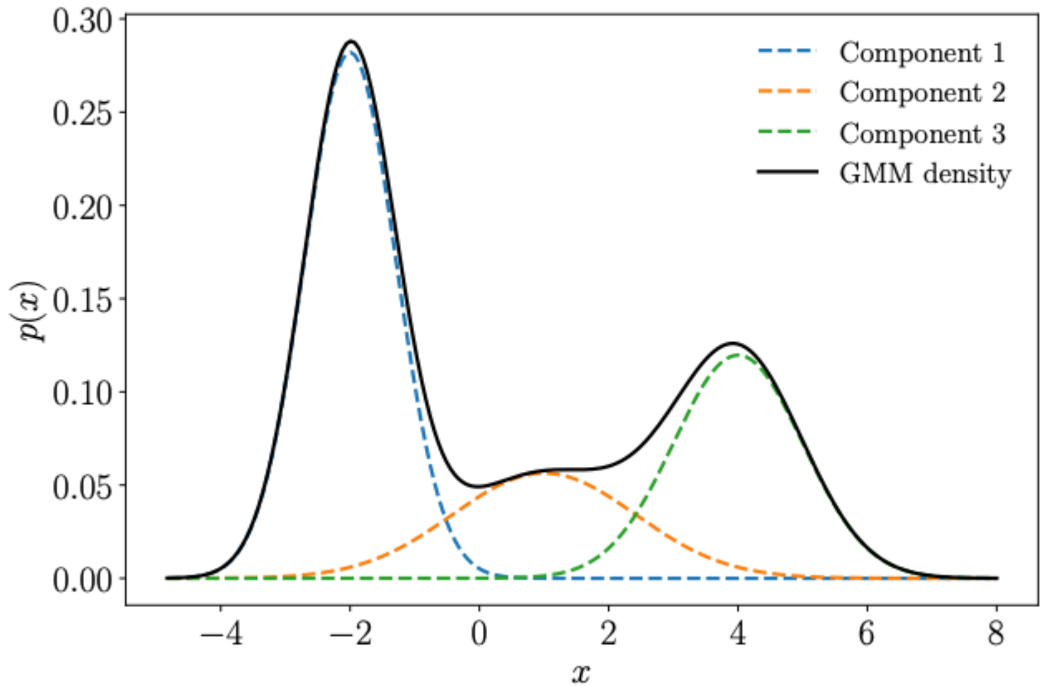
- We can use mixture models for density estimation.
- Mixture models can be used to describe a distribution $p(\boldsymbol{x})$ by a convex combination of K simple (base) distributions

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k p_k(\boldsymbol{x})$$

$$0 \leq \pi_k \leq 1, \qquad \sum_{k=1}^{K} \pi_k = 1$$

where the components p_k are members of a family of basic distributions, e.g., Gaussians, Bernoullis, or Gammas, and the π_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.

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

11.1 Gaussian Mixture Model

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

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_\mathrm{k})$$

$$0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1$$

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

- GMM gives us significantly more flexibility for modeling complex densities than a simple Gaussian distribution.

11.2 Parameter Learning via Maximum Likelihood

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

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{\mathcal{X}})}_{\cancel{\text{posterior}}} \qquad\qquad \overbrace{p(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta})}^{\text{likelihood}}$$

What we want to get | Maximum likelihood optimization

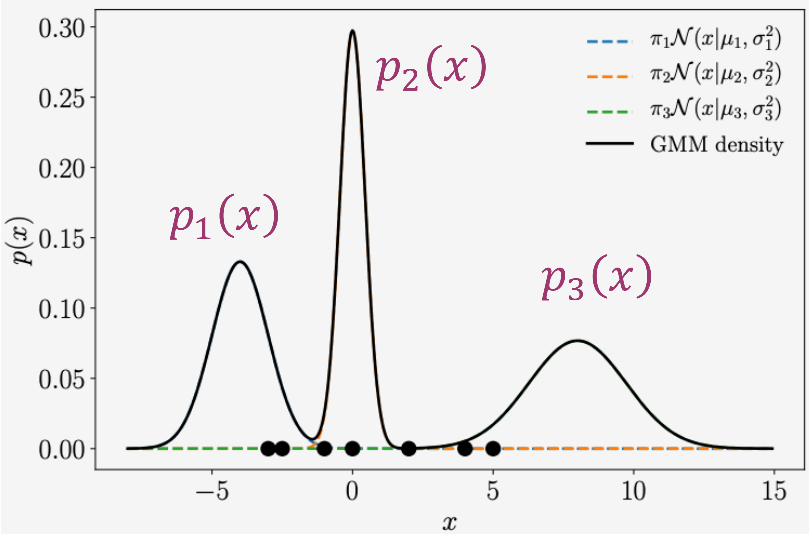
Example

- 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

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

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

$$p(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(\boldsymbol{x}_n|\boldsymbol{\theta}), \qquad p(\underbrace{\boxed{\boldsymbol{x}_n}}_{\text{Observed data}}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \underbrace{\boxed{\pi_k}}_{\text{Mixture proportion}} \underbrace{\boxed{\mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{\mathrm{k}}\right)}}_{\text{Mixture component}}$$

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

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

$$\mathcal{L}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k) = \log p(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

- We aim to find parameters $\boldsymbol{\theta}_{ML}^{*}$ (including $\boldsymbol{\mu}_{\boldsymbol{k}}^{*}, \boldsymbol{\Sigma}_{\boldsymbol{k}}^{*}, \pi_{\boldsymbol{k}}^{*}$) 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 $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{\mathrm{k}}, \pi_k$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}_k} = \mathbf{0}^{\mathrm{T}} \Leftrightarrow \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \mathbf{0}^{\mathrm{T}}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}_k} = 0 \Leftrightarrow \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = \mathbf{0}$$

$$\frac{\partial \mathcal{L}}{\partial \pi_k} = 0 \Leftrightarrow \sum_{n=1} \frac{\partial \log p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \pi_k} = 0$$

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

$$\frac{\partial \log p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{1}{p(\boldsymbol{x}_n|\boldsymbol{\theta})} \frac{\partial p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

- where $\boldsymbol{\theta} = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{\mathrm{k}}, \pi_k \colon k = 1, \cdots, K\}$ are the model parameters and

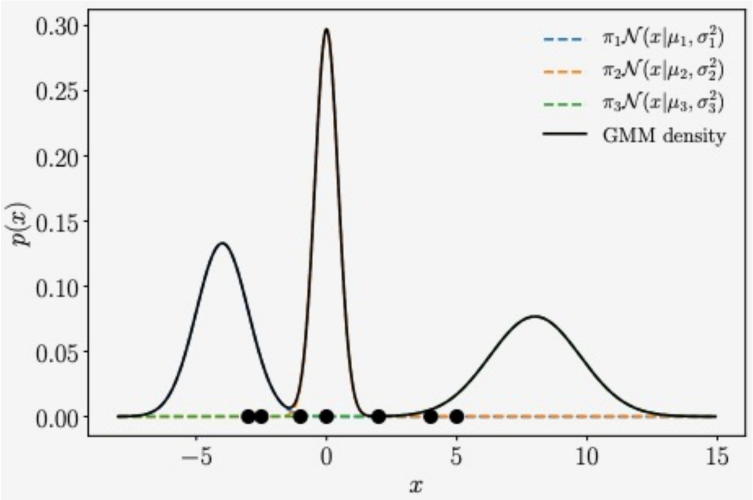
$$\frac{1}{p(\boldsymbol{x}_n|\boldsymbol{\theta})} = \frac{1}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\boldsymbol{x}_n \,\middle|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$$

11.2.1 Responsibilities

- We define the quantity
$$r_{nk} := \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$$
as the responsibility of the kth mixture component for the nth data point.
- We can see r_{nk} is proportional to the likelihood
$$p\left(\boldsymbol{x}_n \middle| \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) = \pi_k \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$
of the kth mixture component given the data point $\boldsymbol{x}_n$.
- The responsibility r_{nk} represents the posterior probability that $\boldsymbol{x}_n$ has been generated by the kth mixture component
- Note that $\boldsymbol{r}_n := [r_{n1}, \cdots, r_{nK}]^{\mathrm{T}} \in \mathbb{R}^K$ 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 $\boldsymbol{r}_n$ as a "soft assignment" of $\boldsymbol{x}_n$ to the K mixture components.

Responsibilities - example

- From the figure below, suppose we have computed the responsibilities r_{nk}



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

- The nth 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 kth column gives us an overview of the responsibility of the kth 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 kth 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 $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and π_k for given responsibilities

11.3 EM Algorithm

- In GMM, we first initialize the parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and π_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 $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and π_k

11.3 EM Algorithm

- Initialize $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k$. (below is an example)
  - $\pi_k = 1/K$ for all k
  - $\boldsymbol{\mu}_k$: centroids from k-means algorithm or using randomly chosen data points
  - $= \boldsymbol{\Sigma}$ the sample variance, for all k

- E-step: Evaluate responsibilities r_{nk} for every data point $\boldsymbol{x}_n$ using current parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$:

$$r_{nk} = \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_j \pi_j \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}$$

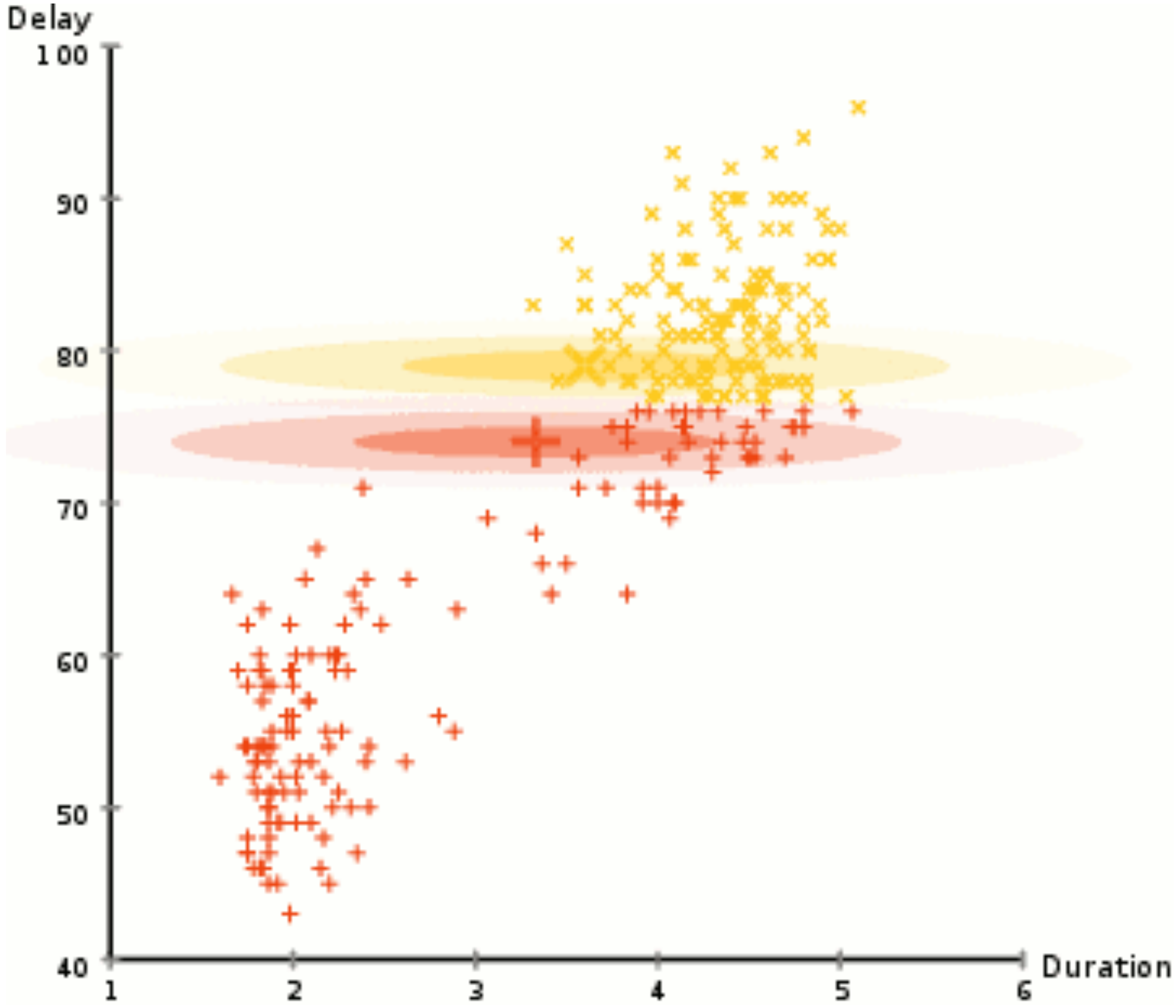
- M-step: Re-estimate parameters $\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ using the current responsibilities r_{nk} (from E-step):

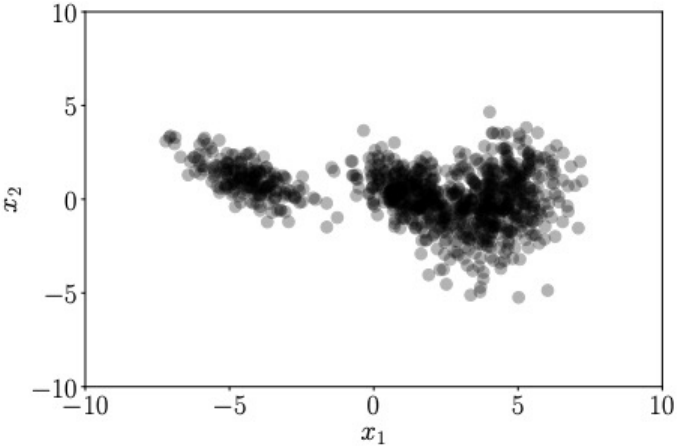
$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n$$

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

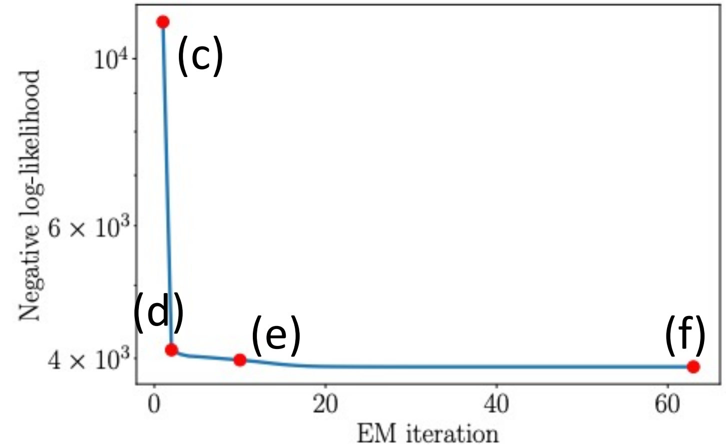
$$\pi_k = \frac{N_k}{N}$$

11.3 EM Algorithm

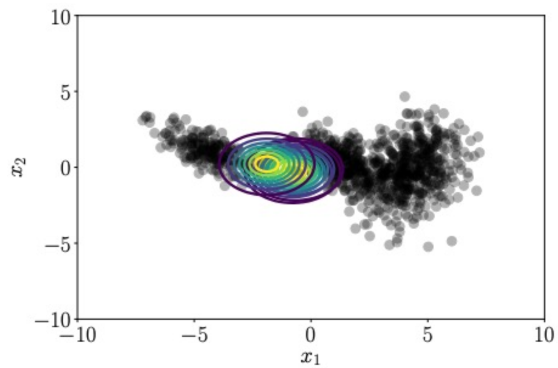




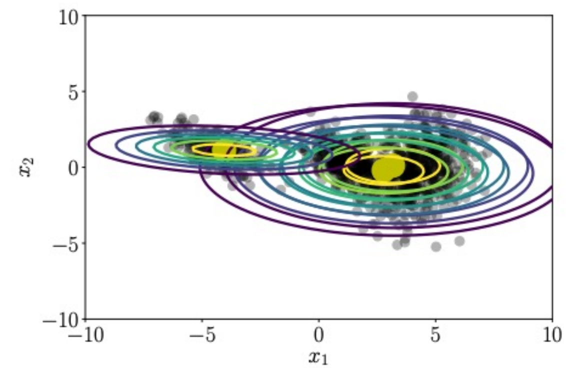
(a) Dataset.



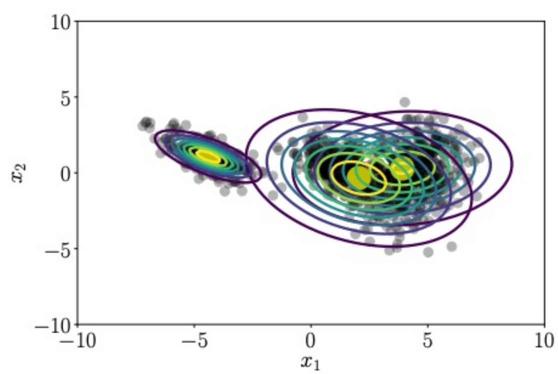
(b) Negative log-likelihood.



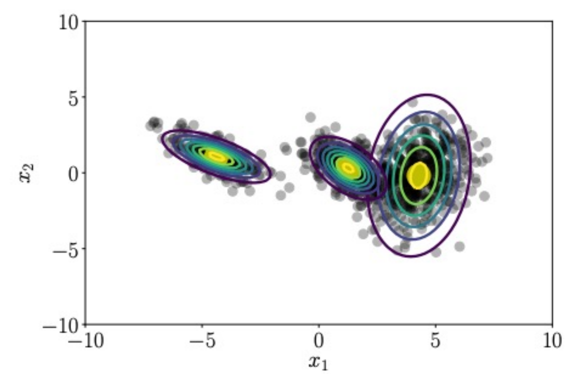
(c) EM initialization.



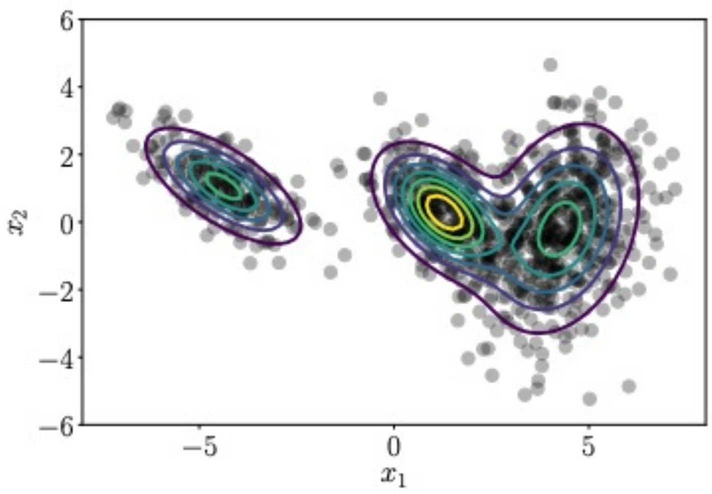
(d) EM after one iteration.



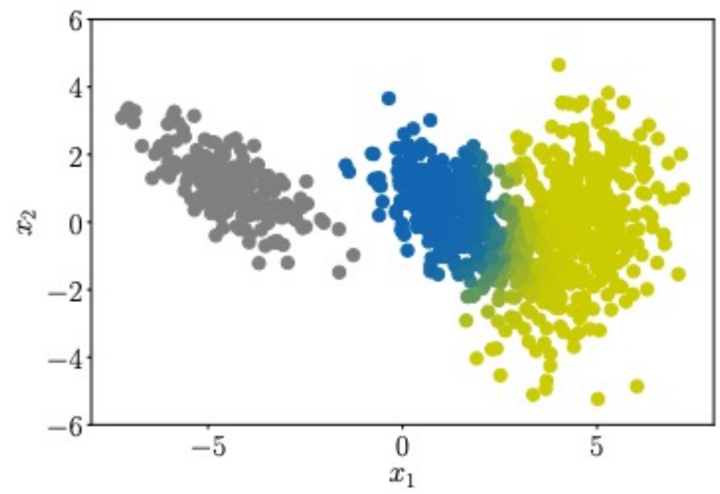
(e) EM after 10 iterations.



(f) EM after 62 iterations.



(a) GMM fit after 62 iterations.



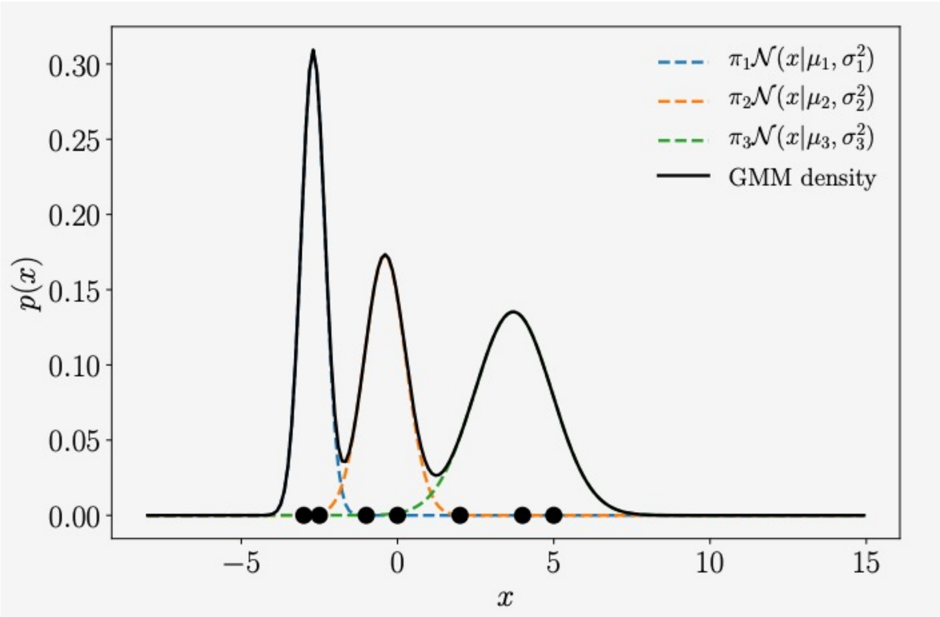
(b) Dataset colored according to the responsibilities of the mixture components.

- 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.

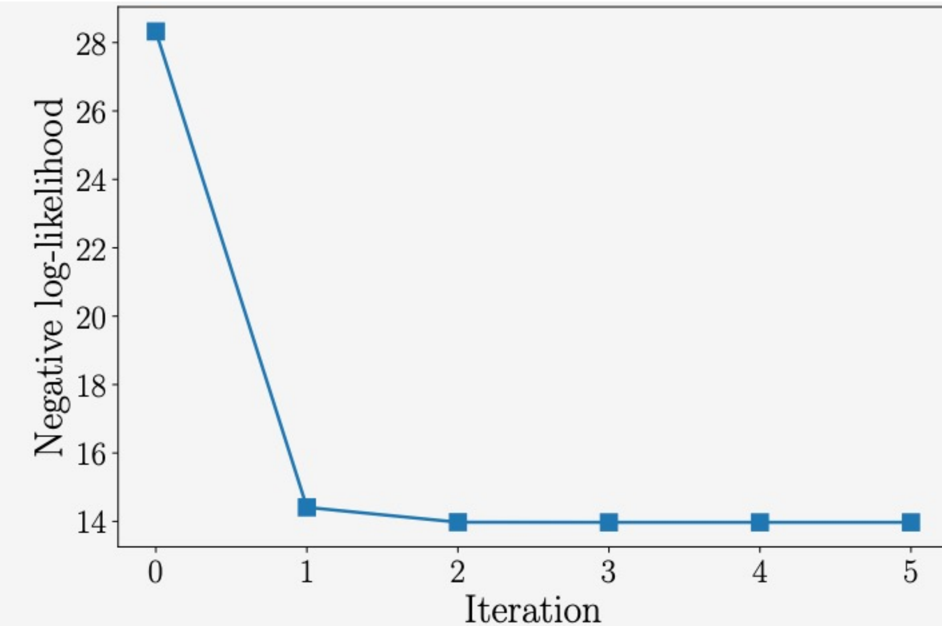
11.3 EM Algorithm

- 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.

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.
- Maximum likelihood optimization comes from Bayes' theory.

$p(\boldsymbol{\theta}|\boldsymbol{\mathcal{X}})$ $\qquad$ $p(\boldsymbol{\mathcal{X}}|\boldsymbol{\theta})$

11.2.2 Updating the Means

- The update of the mean parameters $\boldsymbol{\mu}_k, k = 1, \ldots, K$, of the GMM is given by

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

- Proof: Calculate the gradient of the log-likelihood with respect to $\boldsymbol{\mu}_k$
- Considering

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

$$p(\boldsymbol{x}_n|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_\mathrm{k}\right)$$

- We have

$$\frac{\partial p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_{\boldsymbol{j}}\right)}{\partial \boldsymbol{\mu}_k} = \pi_k \frac{\partial \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\partial \boldsymbol{\mu}_k}$$

- Recall our knowledge in multivariate Gaussian distribution and vector calculus

$$p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{D}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}} exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}))$$

$$\frac{\partial \boldsymbol{x}^\mathrm{T}\boldsymbol{B}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{x}^\mathrm{T}(\boldsymbol{B} + \boldsymbol{B}^\mathrm{T})$$

- We have

$$\frac{\partial p(\boldsymbol{x}_n|\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \pi_k(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^\mathrm{T}\boldsymbol{\Sigma}_k^{-1}\mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$$

11.2.2 Updating the Means

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

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

$$= \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1} \underbrace{\boxed{\frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}}}_{= r_{nk}}$$

$$= \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1}$$

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

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

where we define

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

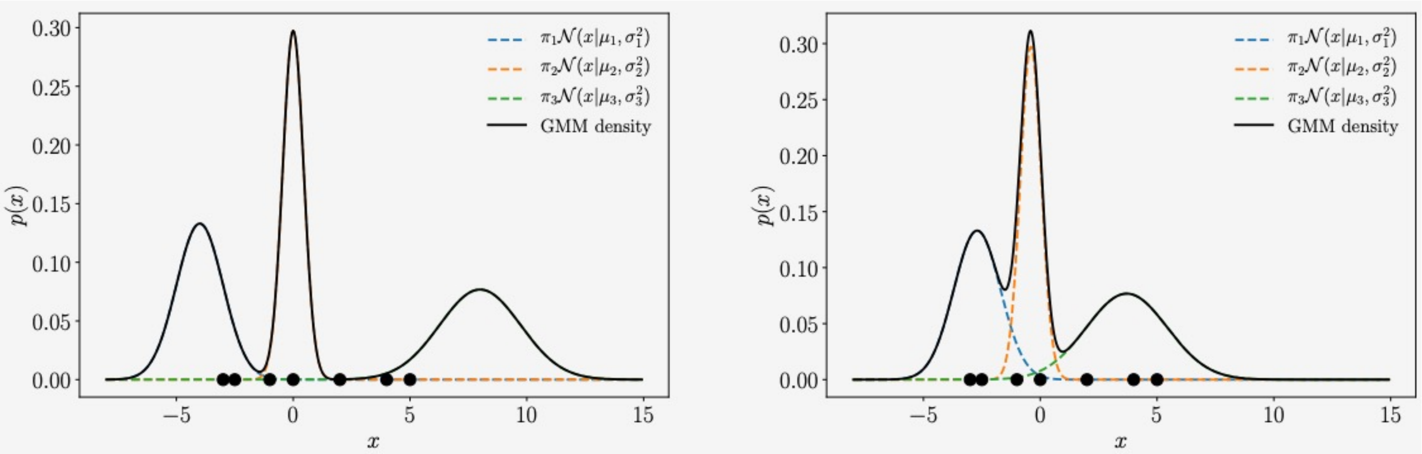
as the total responsibility of the *k*th mixture component for the entire dataset.

- This concludes the proof.

11.2.2 Updating the Means

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

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



Initialization:

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

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

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

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

$$\mu_1 : -4 \longrightarrow -2.7$$
$$\mu_2 : 0 \longrightarrow -0.4$$
$$\mu_3 : 8 \longrightarrow 3.7$$

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

11.2.3 Updating the Covariances

- The update of the covariance parameters $\boldsymbol{\Sigma}_k, k = 1, \ldots, K$ is given by

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

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

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

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

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

11.2.3 Updating the Covariances

- From Vector Calculus, we have the following identities

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_k} \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} = -\frac{1}{2} \det(\boldsymbol{\Sigma}_k)^{-\frac{1}{2}} \boldsymbol{\Sigma}_k^{-1}$$

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

- We obtain the desired partial derivative

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

- Thus, the partial derivative of the log-likelihood with respect to $\boldsymbol{\Sigma}_k$ is given by

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

$$= \sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}}_{= r_{nk}} \cdot \left[-\frac{1}{2}\left(\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1}\right)\right]$$

$$= -\frac{1}{2} \sum_{n=1}^{N} r_{nk} \left(\boldsymbol{\Sigma}_k^{-1} - \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1}\right)$$

$$= -\frac{1}{2} \boldsymbol{\Sigma}_k^{-1} \underbrace{\sum_{n=1}^{N} r_{nk}}_{N_k} + \frac{1}{2} \boldsymbol{\Sigma}_k^{-1} \left(\sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}\right) \boldsymbol{\Sigma}_k^{-1}$$

11.2.3 Updating the Covariances

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

$$N_k \boldsymbol{\Sigma}_k^{-1} = \boldsymbol{\Sigma}_k^{-1} \left(\sum_{n=1}^{N} r_{nk} \, (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}} \right) \boldsymbol{\Sigma}_k^{-1}$$

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

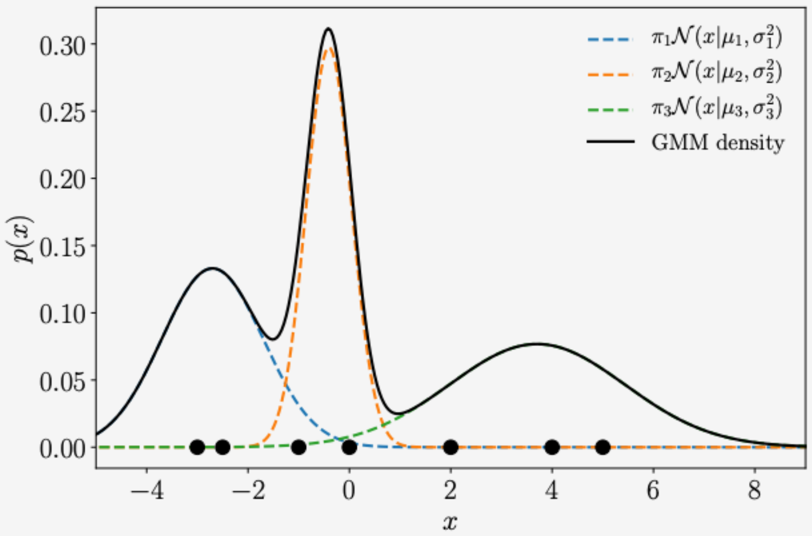
- By solving for $\boldsymbol{\Sigma}_k$, we obtain

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

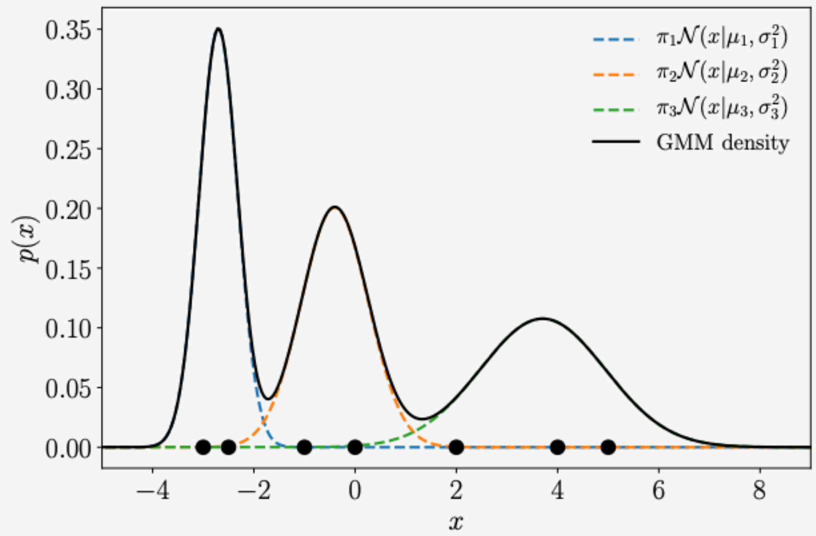
- This gives us a simple update rule for $\boldsymbol{\Sigma}_k$ for $k = 1, \dots, K$ and proves our theorem.

- This update method is the weighted covariance of data points $\boldsymbol{x}_n$ associated with the kth 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.

$$\sigma_1^2 : 1 \longrightarrow 0.14$$
$$\sigma_2^2 : 0.2 \longrightarrow 0.44$$
$$\sigma_3^2 : 3 \longrightarrow 1.53$$

11.2.4 Updating the Mixture Weights

- The mixture weights of the GMM are updated as

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

where N is the number of data points

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

$$\sum_k \pi_k = 1$$

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

$$\mathfrak{L} = \mathcal{L} + \lambda \left(\sum_{k=1}^{K} \pi_k - \mathbf{1} \right)$$

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

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

- We obtain the partial derivative with respect to π_k as

$$\frac{\partial \mathfrak{L}}{\partial \pi_k} = \sum_{n=1}^{N} \frac{\mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)} + \lambda$$

$$= \frac{1}{\pi_k} \underbrace{\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j \mathcal{N}\left(\boldsymbol{x}_n \middle| \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}}_{= N_k} + \lambda = \frac{N_k}{\pi_k} + \lambda$$

- The partial derivative with respect to the Lagrange multiplier λ is

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

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

$$\begin{cases} \pi_k = -\dfrac{N_k}{\lambda} \\ 1 = \displaystyle\sum_{k=1}^{K} \pi_k \end{cases}$$

- Using the two equations, we obtain

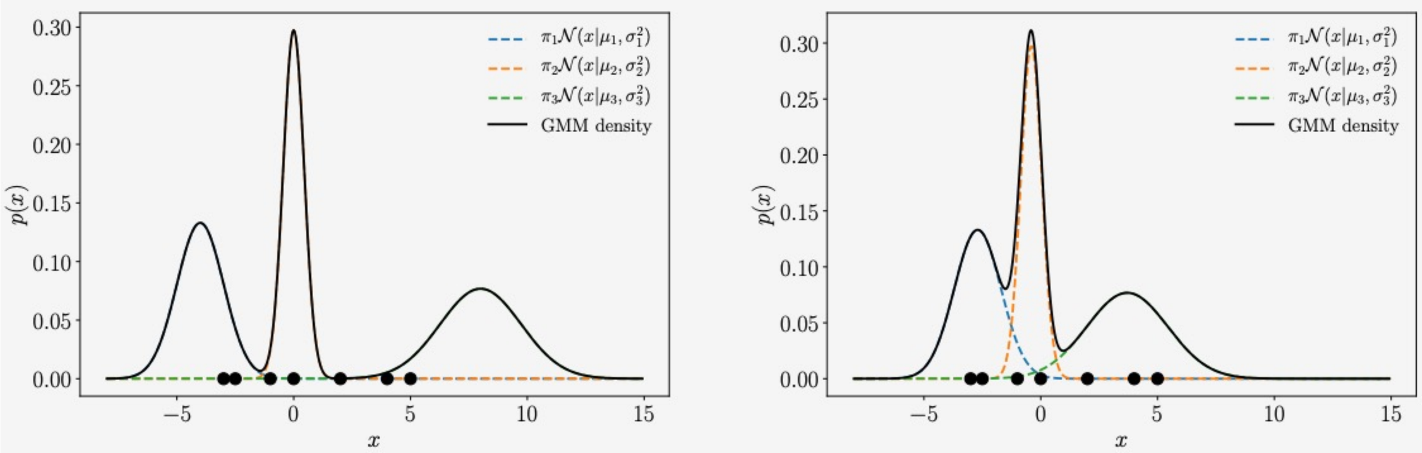
$$\sum_{k=1}^{K} \pi_k = 1 \Leftrightarrow -\sum_{k=1}^{K} \frac{N_k}{\lambda} = 1 \Leftrightarrow -\frac{N}{\lambda} = 1 \Leftrightarrow \lambda = -N$$

$$\boxed{\begin{cases} \pi_k = -\dfrac{N_k}{\lambda} \\ 1 = \sum_{k=1}^{K} \pi_k \end{cases}}$$

- This allows us to substitute $-N$ for λ 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 π_k and proves the Theorem.



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

$$\pi_1 : \frac{1}{3} \longrightarrow 0.29$$

$$\pi_2 : \frac{1}{3} \longrightarrow 0.29$$

$$\pi_3 : \frac{1}{3} \longrightarrow 0.42$$

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

- 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 $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k, k = 1, \dots, K$, we want to generate a dataset with N data points.
- ~~We sample an index~~ $\cancel{k}$ ~~from~~ $\cancel{\{1, 2, \dots, K\}}$ ~~with probabilities~~ $\cancel{\pi_1, \dots, \pi_K}$

- We generate a number of $N\pi_k$ data points for the kth component
- In the kth component, every data point is sampled as $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Comparing 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 π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ 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.