Expectation Maximization for Gaussian Mixture Models and Hidden Mixture Models

Applications to speech and further consideration

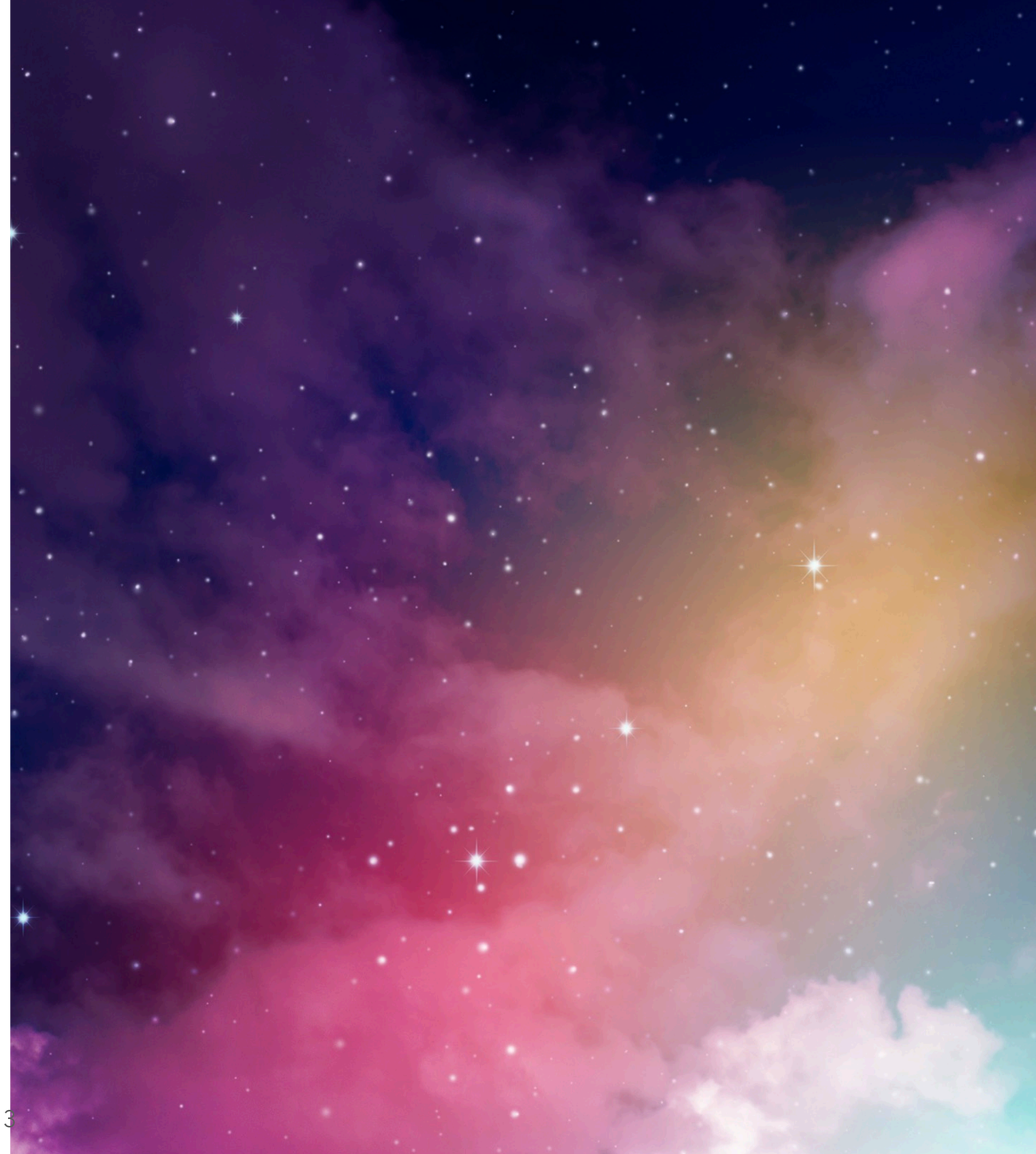
EPFL

Juan Pablo Zuluaga, Mael Fabien - 07 May 2020

Overview

1. Modeling distributions with k-Means and GMMs
2. Reminder on GMMs
3. Motivation for EM
4. EM for GMMs
5. Extensions and special cases of EM
6. EM for HMMs
7. EM for HMM/GMM
8. Other approaches to distribution modeling

I.
Modeling distributions with k-Means and GMMs



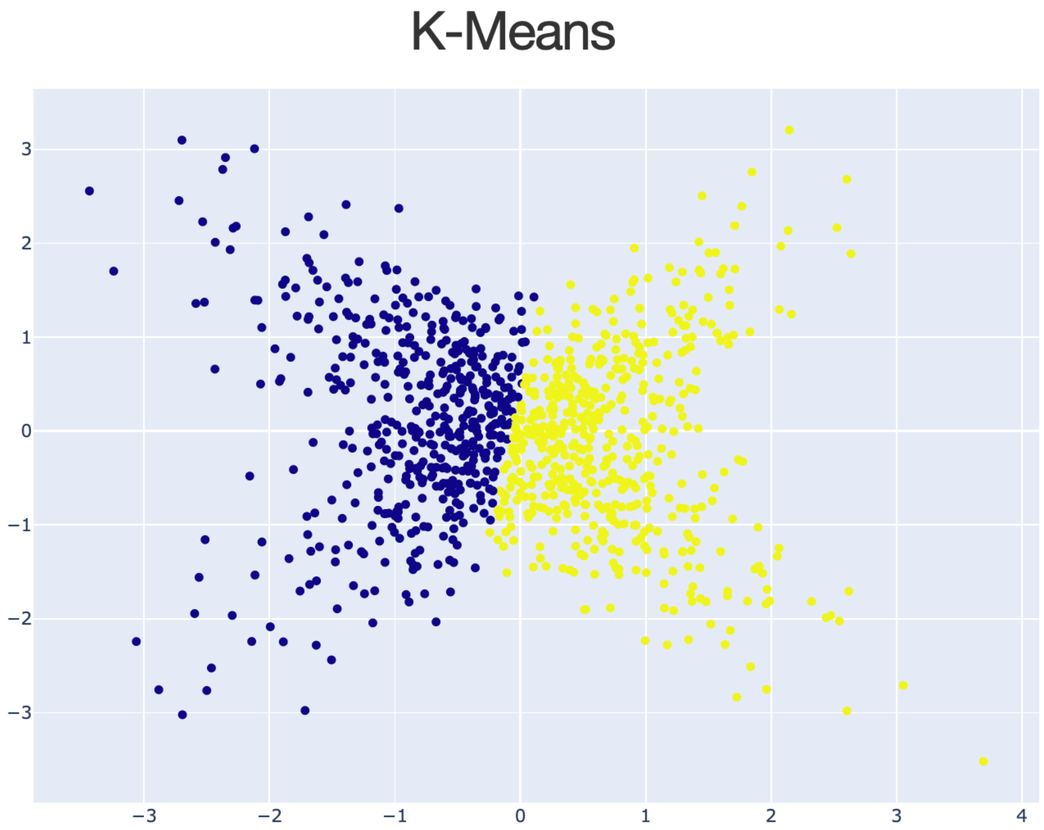
I. Modeling distributions with k-Means and GMMs

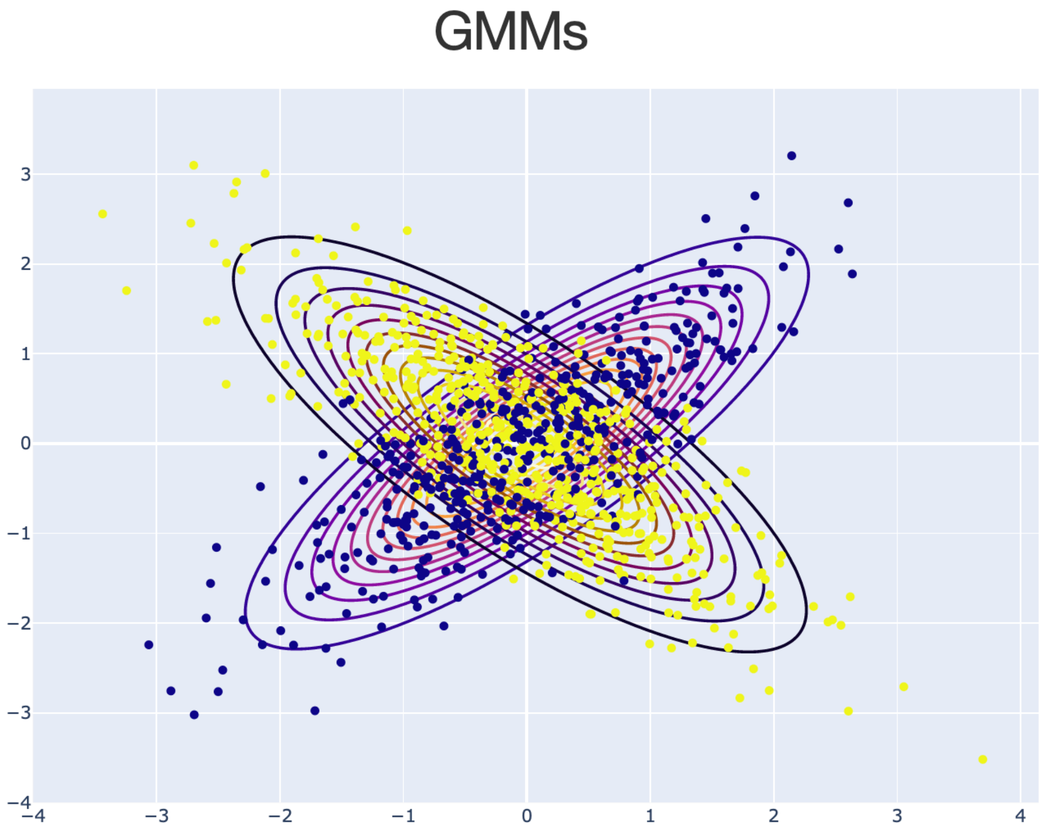
- Why are GMMs a popular choice for modeling distributions? And how does it compare to k-Means? GMMs can be seen as extensions of k-Means

k-Means	GMM
Clusters are defined by their means	Clusters are defined by their means and their variance, modeled as Gaussians
Does not work If clusters are overlapping	Works if clusters are overlapping
Uses Euclidean distance to the mean	Uses the probability of X belonging to a cluster (generative)

I. Modeling distributions with k-Means and GMMs

- Why are GMMs a popular choice for modeling distributions? And how does it compare to k-Means?





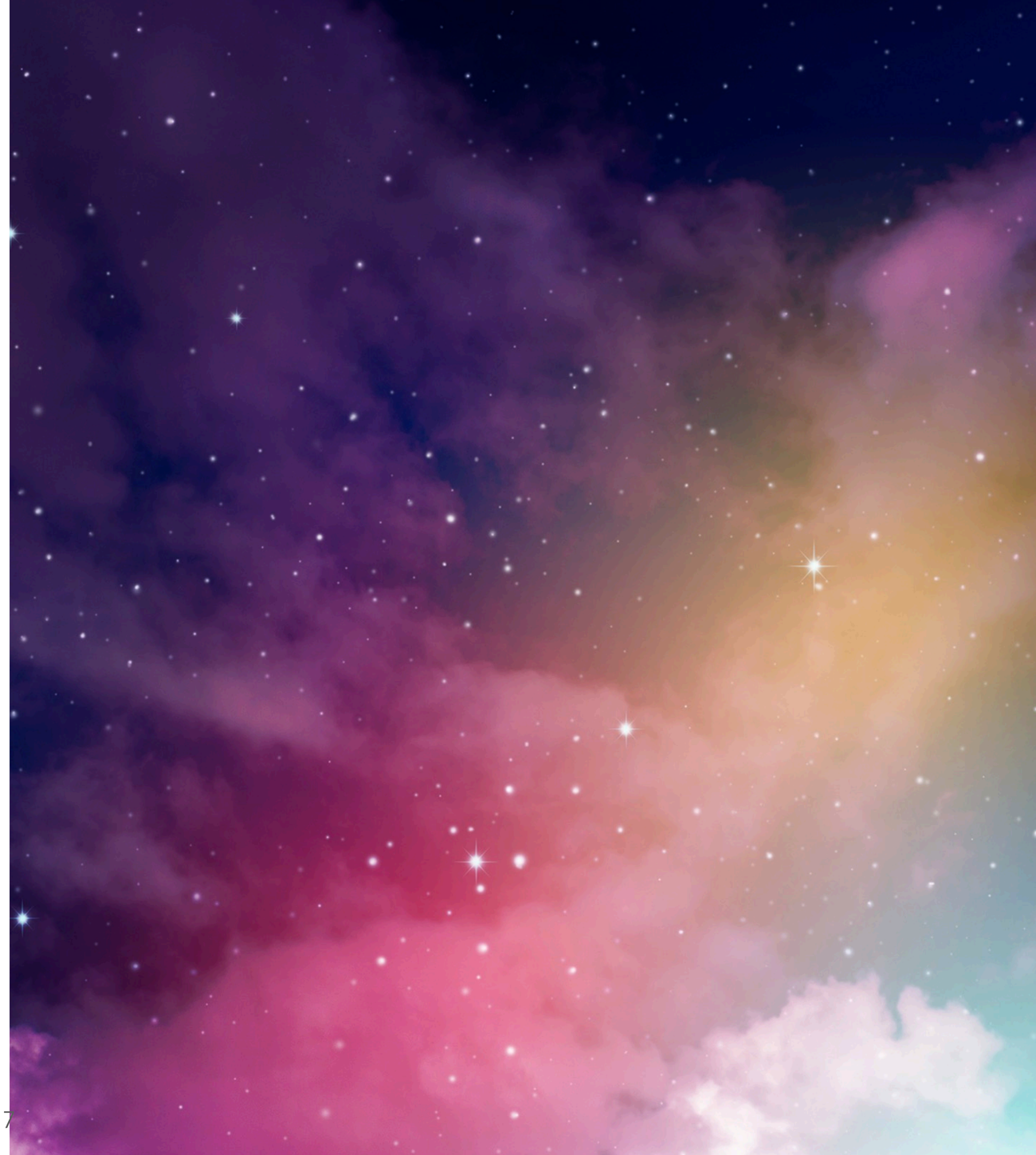
I. Modeling distributions with k-Means and GMMs

- Why are GMMs a popular choice for modeling distributions? And how does it compare to k-Means?

Moreover, GMMs, since generative models, have some useful properties, such as:

- Deriving the probability that observations come from a cluster
- Evaluating the similarity between training and testing sets

II.
Reminder on Gaussian Mixture Models



II. Reminder on Gaussian Mixture Models

- A GMM is a weighted sum of M components Gaussian densities. A density of a Gaussian can be defined as:

$$P(x \mid \lambda) = \sum_{k=1}^{M} w_k \, \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

M Components

Weights

Gaussian density

$$\sum_{k=1}^{M} w_k = 1$$

$$\mathcal{N}(x \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}} \mid \Sigma_k \mid^{\frac{1}{2}}} exp^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}$$

II. Reminder on Gaussian Mixture Models

```python
from scipy.stats import multivariate_normal


def make_data(n_data, means, covariances, weights):

    n_clusters, n_features = means.shape
    list_clusters = []

    data = np.zeros((n_data, n_features))
    for i in range(n_data):

        k = np.random.choice(n_clusters, size = 1, p = weights)[0]
        list_clusters.append(k)

        x = np.random.multivariate_normal(means[k], covariances[k])
        data[i] = x

    return data, list_clusters
```

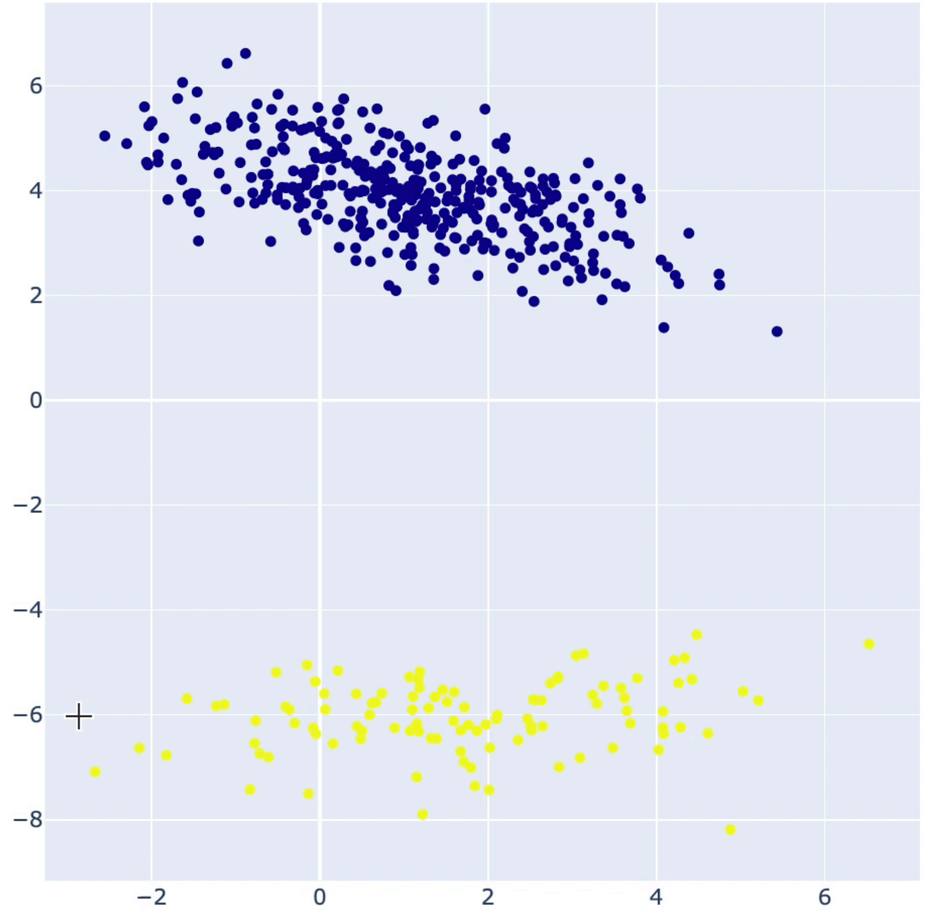
II. Reminder on Gaussian Mixture Models

Parameters

Generating data from GMMs

Number of components

1 2 3 4 5 6 7 8



II. Reminder on Gaussian Mixture Models

- The parameters of the GMM are therefore : $\lambda = (w_k, \mu_k, \Sigma_k),\ k = \ 1,\ 2,\ 3,\ \ldots,\ M$

  How do we solve GMM? Start by solving a ***single gaussian***…

- We apply a Maximum Likelihood Estimation (MLE) to find the parameters:

$$L(\theta \mid X) = \prod_{i=1}^{N} P(x_i \mid \theta) = \prod_{i=1}^{N} \frac{1}{\sqrt{(2\pi\sigma^2)}} exp^{\frac{-(x_i-\mu)^2}{2\sigma^2}} \qquad \theta = (\mu, \sigma)$$

$$\theta^\star = argmax_\theta\ L(\theta \mid X)$$

II. Reminder on Gaussian Mixture Models

- For convenience, we maximize the log-likelihood since:

$$argmax_\theta \; L(\theta \mid X) = argmax_\theta \; \log L(\theta \mid X)$$

- Set the derivatives for both parameters to 0:

$$\frac{d}{d\mu} logL(\theta \mid X) = 0 \quad \dashrightarrow \quad \mu_{MLE} = \frac{1}{N}\sum_{n=1}^{N} x_N$$

$$\frac{d}{d\sigma} logL(\theta \mid X) = 0 \quad \dashrightarrow \quad \sigma^2_{MLE} = \frac{1}{N}\sum_{n=1}^{N} (x_N - \mu)^2$$

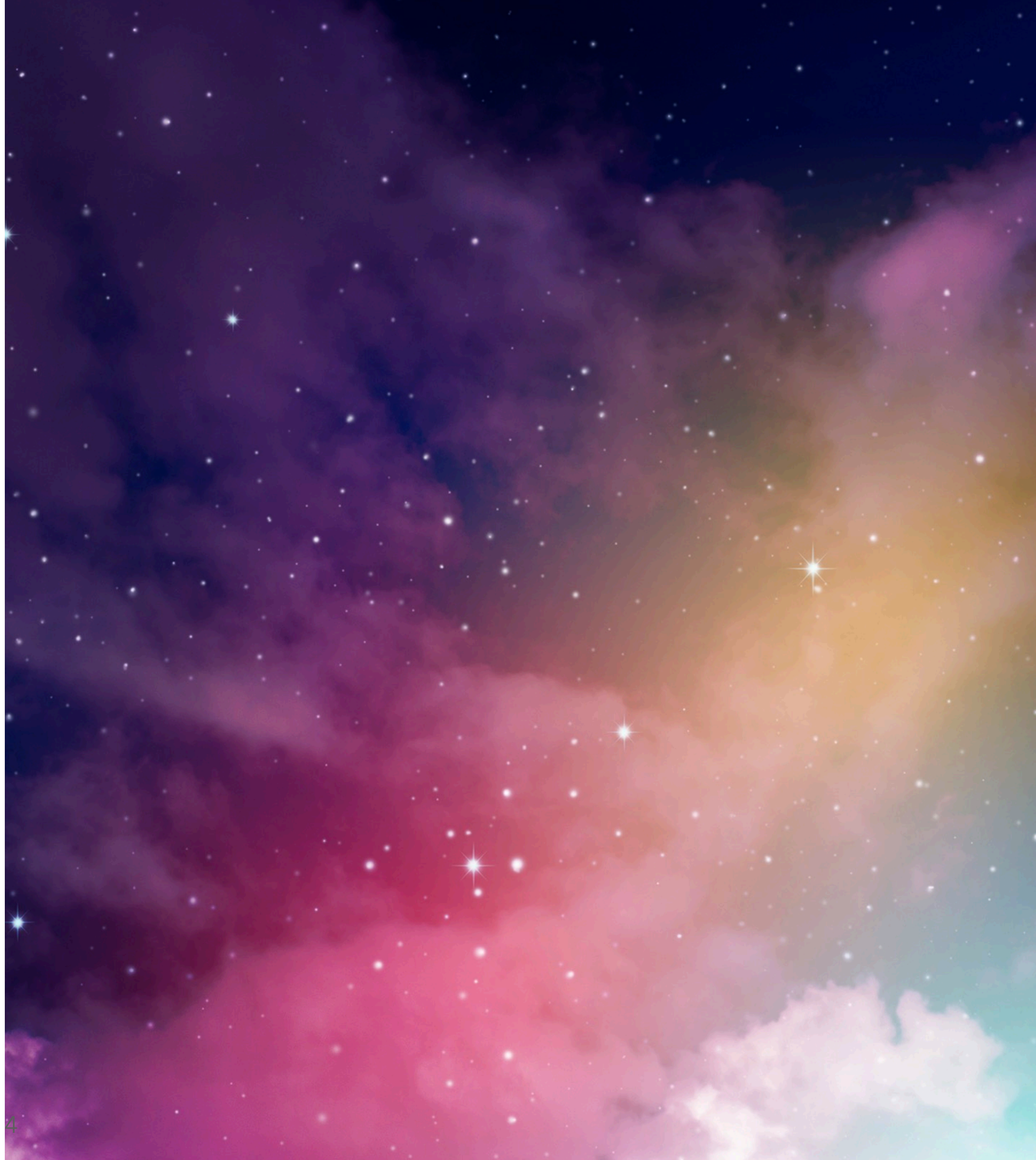
II. Reminder on Gaussian Mixture Models

- To find the MLE of GMM parameters, we need to re-define the likelihood:

$$L(\theta \mid X_1, \ldots, X_n) = \prod_{i=1}^{N} \sum_{k=1}^{M} w_k \, \mathcal{N}(x_i; \mu_k; \sigma_k^2)$$

All observations — All components — Component weights — Gaussians

III.
Motivation for Expectation Maximization



III. Motivation for EM

- The log-likelihood becomes:

$$l(\theta) = \log L(\theta \mid X_1, \ldots, X_n) = \sum_{i=1}^{N} \log(\sum_{k=1}^{M} w_k \mathcal{N}(x_i, \mu_k, \sigma_k^2))$$

- We now must solve over the M Gaussian components. If we set the derivative to 0 to identify the optimal value of the means μ_k :

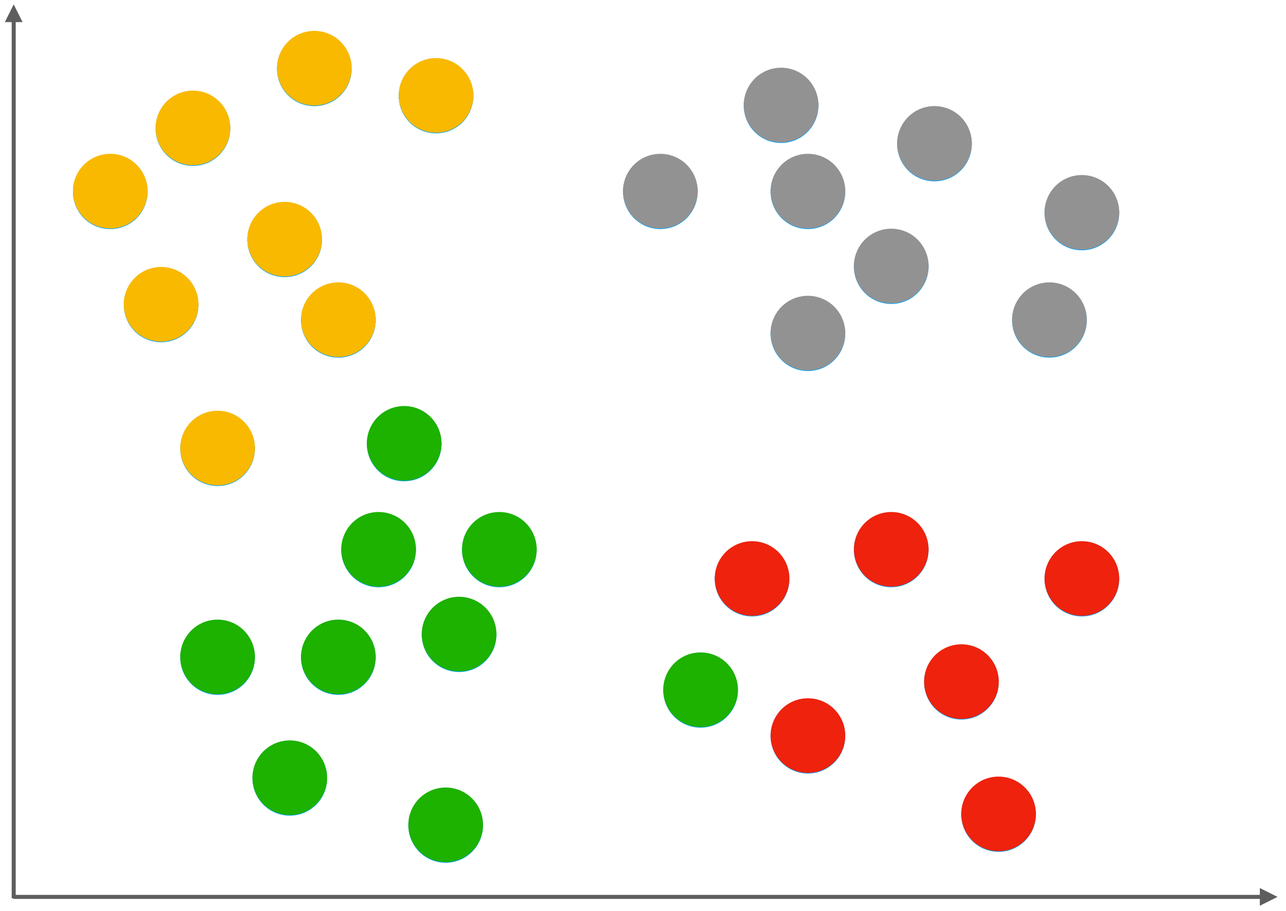
$$\sum_{i=1}^{N} \frac{1}{\sum_{k=1}^{M} w_k \mathcal{N}(x_i, \mu_k, \sigma_k)} w_k \mathcal{N}(x_i, \mu_k, \sigma_k) \frac{(x_i - \mu_k)}{\sigma_k^2} = 0$$

III. Motivation for EM

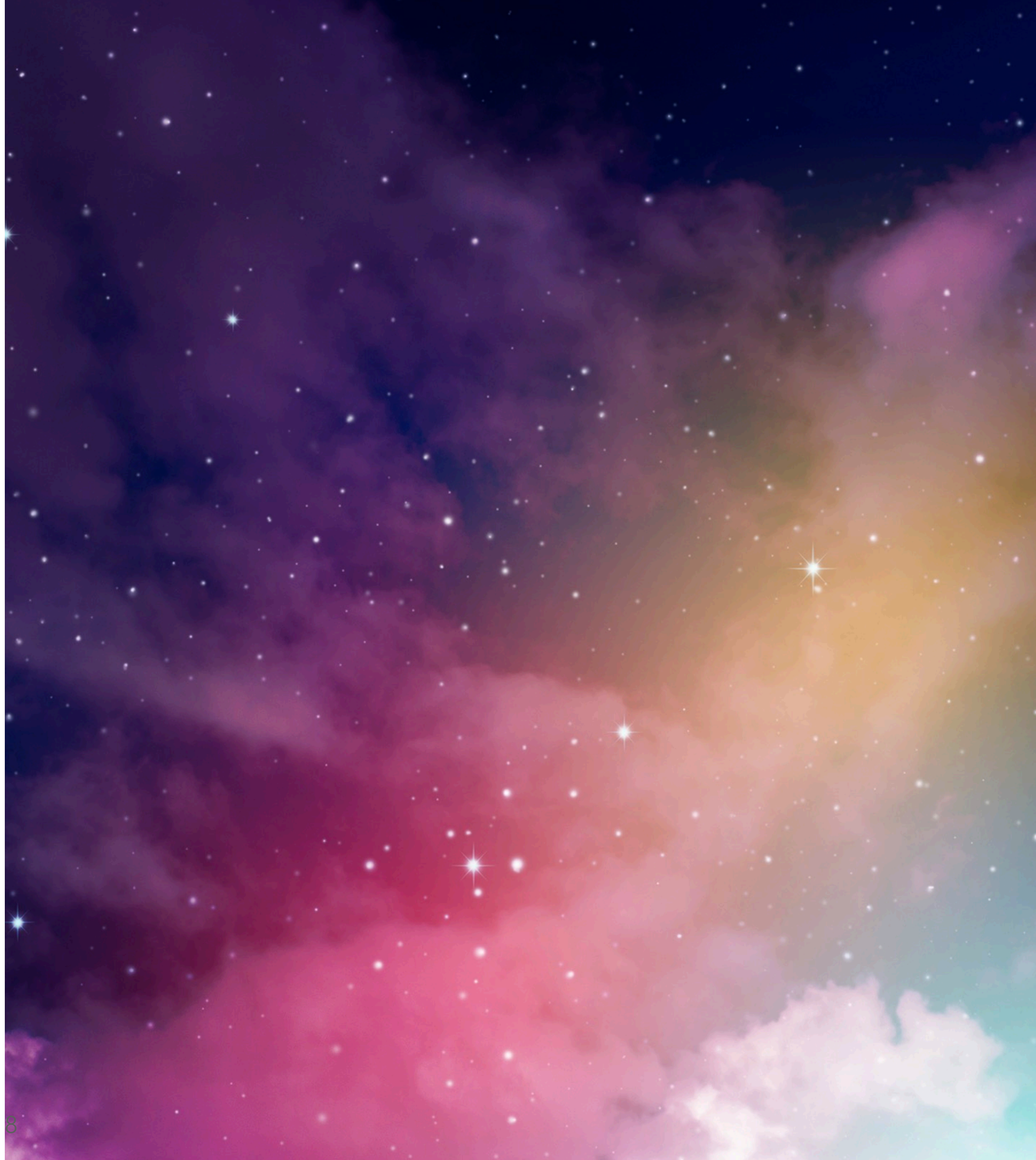
- We cannot solve analytically for μ_k , and therefore we need to find another approach
- We can find the parameters of single Gaussians using MLE, so what if we knew which Gaussian each data point belongs to? We could solve analytically for the parameters of each Gaussian
- We suppose that latent variables Z_i exist, and they describe the component of the mixture to which X_i belongs

III. Motivation for EM

- We can suppose that we know which component each observation X_i belongs to ($Z_i = k$) and it will help us solve each Gaussian. We can iteratively update the parameters using EM in the training cycle.



IV.
EM for Gaussian Mixture Models



IV. EM for GMMs

- EM introduces a latent variable Z corresponding to the component of the GMM to which each observation belongs
- X is now said to be *incomplete data*, and the **complete data** is: (X, Z)
- The joint density is: $P(X, Z \mid \theta) = P(Z \mid X, \theta)P(X \mid \theta)$
- The likelihood $L(\theta \mid X)$ is now said to be *incomplete*
- The **complete likelihood** now becomes : $L(\theta \mid X, Z) = P(X, Z \mid \theta)$

IV. EM for GMMs

- Further expanding the complete log-likelihood, we get:

$$\log(L(\theta \mid X, Z)) = \log(\sum_{Z} P(X, Z \mid \theta)) = \sum_{i=1}^{N} \log P(x_i \mid z_i) P(z_i)$$

- We first initialize the parameters of our Gaussians randomly:

$$\theta = (\mu, \sigma, w)$$

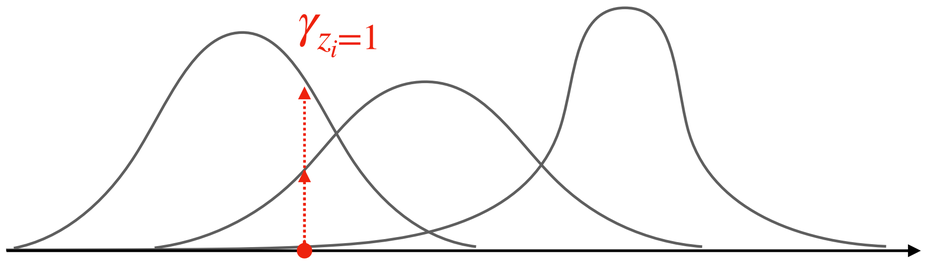
IV. EM for GMMs

- Let us now define $\gamma_{z_i=k}$ as the probability that an observation belongs to cluster k
- These « pseudo-posteriors » are defined as:

$$\gamma_{z_i=k} = P(Z_i = k \mid X_i) = \frac{P(X_i \mid Z_i = k)P(Z_i = k)}{P(X_i)} = \frac{w_k \mathcal{N}(x_i, \mu_k, \sigma_k)}{\sum_c w_c \mathcal{N}(x_i, \mu_c, \sigma_c)}$$

Probability of X under component k

Sum of probabilities on all components



1. The E-Step

- In the « Estimation » step (E-step), we estimate the value of the auxiliary function:

$$Q(\theta,\ \theta^{(t)}) = E[\log P(Z \mid \theta) \mid X, \theta^{(t)}]$$

1. The E-Step

- In the « Estimation » step (E-step), we estimate the value of the auxiliary function:

$$Q(\theta,\ \theta^{(t)}) = \sum_{k=1}^{M} \log L(\theta_k \mid X, Z)\ P(Z_k \mid X, \theta^{(t)}) = \sum_{k=1}^{M} \log L(\theta_k \mid X, Z)\ \gamma_{z_i=k}$$

Likelihood of a Gaussian knowing Z

Priors

Current parameter value

Adjusted parameters

2. The M-Step

- In the « Maximum » step (M-step), we maximize the value of Q to find the optimal parameter value:

$$\theta^{(t+1)} = argmax_{\theta}\ Q(\theta, \theta^{(t)})$$

2. The M-Step

- The update equations of the M-step are defined by setting the derivative of Q to 0 with respect to γ, μ, σ. By expanding the expression of the auxiliary function, we get the following expression which can be solved analytically:

$$Q(\theta, \theta^{(t+1)}) = \underbrace{\sum_{k=1}^{M}\sum_{i=1}^{N} \log \gamma_k P(Z_k \mid X_i, \theta^{(t)})}_{} + \underbrace{\sum_{k=1}^{M}\sum_{i=1}^{N} \log P(x_i \mid \theta_k) P(Z_k \mid X_i, \theta^{(t)})}_{}$$

$$\frac{d}{d\gamma_k}\sum_{k=1}^{M}\sum_{i=1}^{N} \log \gamma_k P(Z_k \mid X_i, \theta^{(t)}) + \underbrace{\lambda(\sum_{k} \gamma_k - 1)}_{} = 0$$

Lagrange multiplier and constraint

2. The M-Step

- The updated parameters become:

$$\gamma_k = \frac{1}{N} \sum_{i=1}^{N} P(Z_i = k \mid X_i, \theta^{(t)})$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} X_i P(Z_i = k \mid X_i, \theta^{(t)})}{\sum_{i=1}^{N} P(Z_i = k) \mid X_i, \theta^{(t)})} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{Z_i=k} X_i \quad \text{where } N_k = \sum_{i=1}^{N} \gamma_{Z_i=k}$$

Weighted average of the data with a weight showing how likely the point belongs to the cluster

$$\hat{\sigma_k^2} = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{Z_i=k} (X_i - \mu_k)^2$$

$$\hat{w}_k = \frac{N_k}{N}$$

3. An iterative process

- Using the new values of the parameters, inject it in the E-step again:

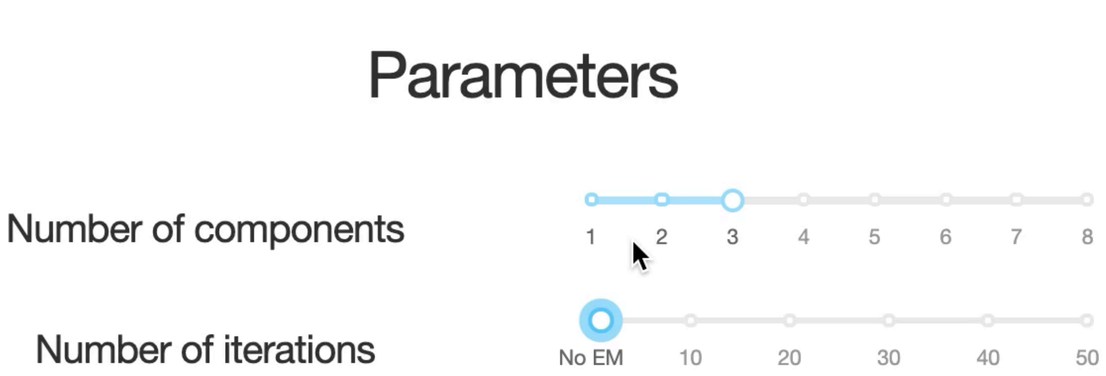
$$Q(\theta, \theta^{(t+1)}) = \sum_{k=1}^{M} \log L(\theta_k \mid X, Z)\ P(Z_k \mid X, \theta^{(t+1)})$$

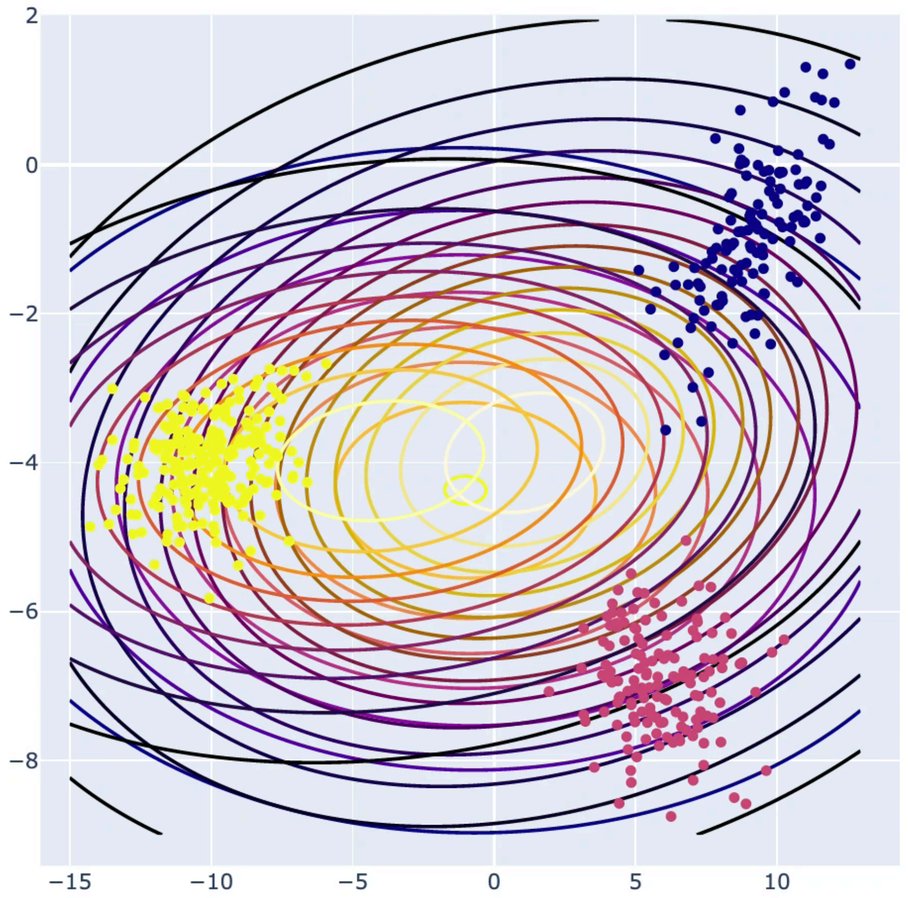
3. An iterative process

- And estimate the value of the optimal parameters again in the M-Step:

$$\theta^{(t+2)} = argmax_\theta \, Q(\theta, \theta^{(t+1)})$$

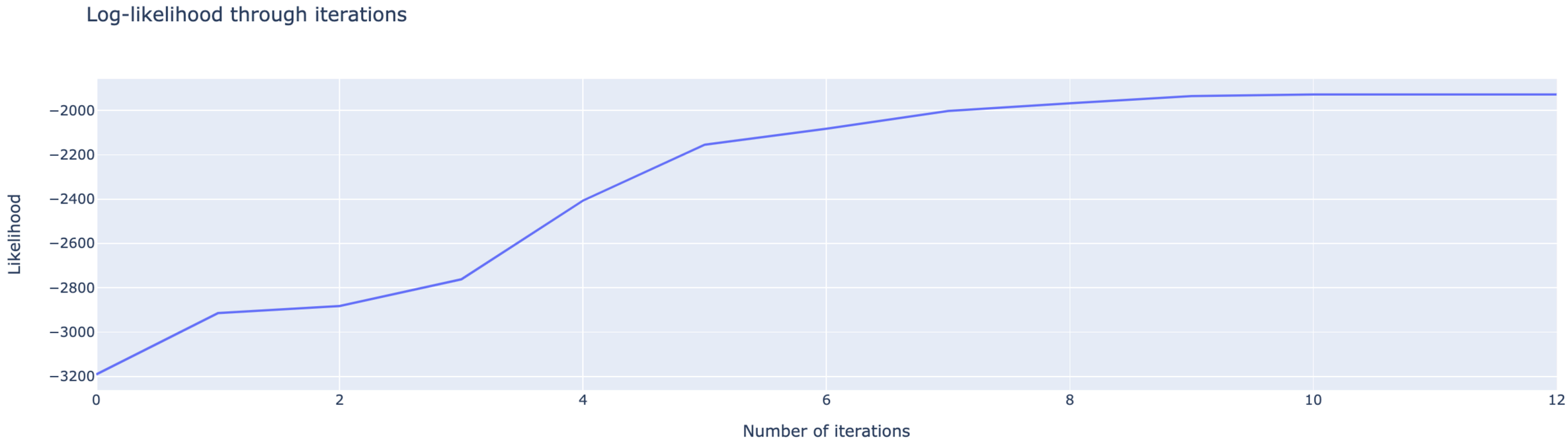
3. An iterative process





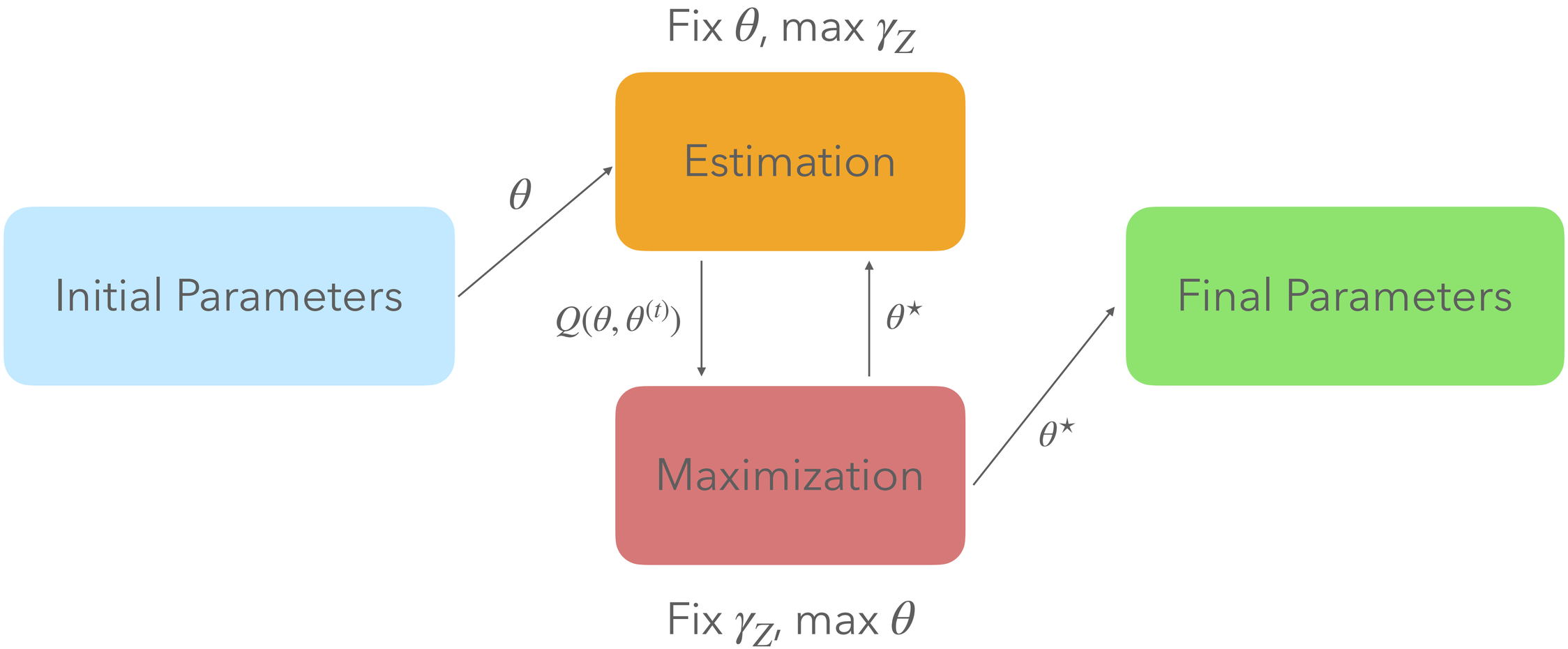
3. An iterative process

- EM is an iterative approach that is guaranteed to increase the likelihood over the number of iterations: $\log p(X \mid \theta^{(t)}) \geq \log p(X \mid \theta)$



3. An iterative process

- The EM cycle can be illustrated as:



4. NB: Where does the auxiliary function come from?

- The origin of the auxiliary function is linked to the Jensen's inequality:

$$L(\theta) - L(\theta^{(t)}) = \log \frac{p(X \mid \theta)}{p(X \mid \theta^{(t)})} = \log \int \frac{p(Z, X \mid \theta)}{p(X \mid \theta^{(t)})} dz$$

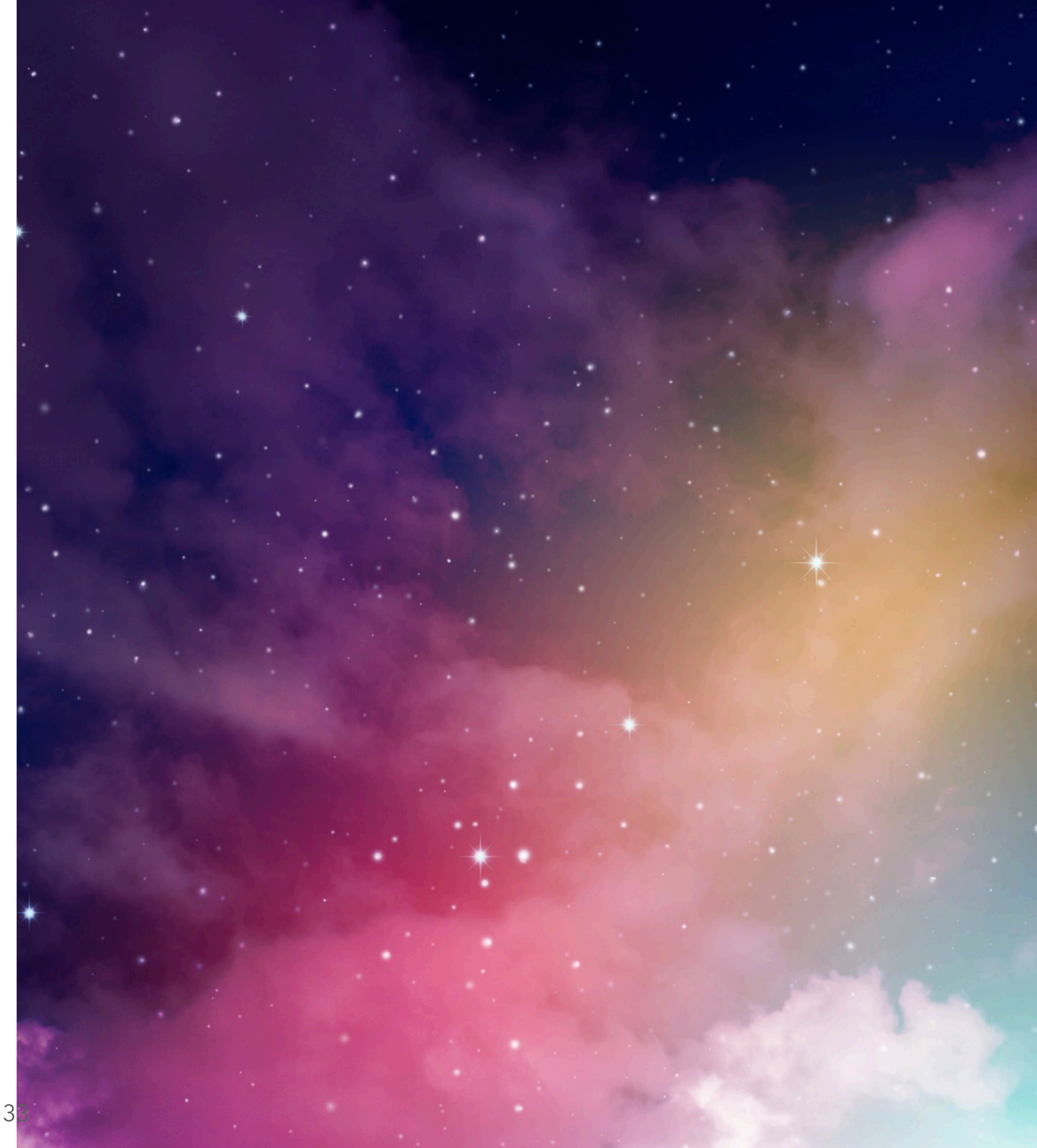
$$= \log \int \frac{p(Z, X \mid \theta)}{p(Z, X \mid \theta^{(t)})} P(Z \mid X, \theta^{(t)}) dz = \log \int \frac{p(Z \mid \theta)}{p(Z \mid \theta^{(t)})} P(Z \mid X, \theta^{(t)}) dx$$

$$\geq \underbrace{\int \log \frac{p(Z \mid \theta)}{p(Z \mid \theta^{(t)})} P(Z \mid X, \theta^{(t)}) dx}_{\text{Auxiliary function } Q(\theta, \theta^{(t)})}$$

Jensen's Inequality

The true likelihood variation is always **greater** than the variation of the auxiliary function

V.
Extensions and special cases of EM



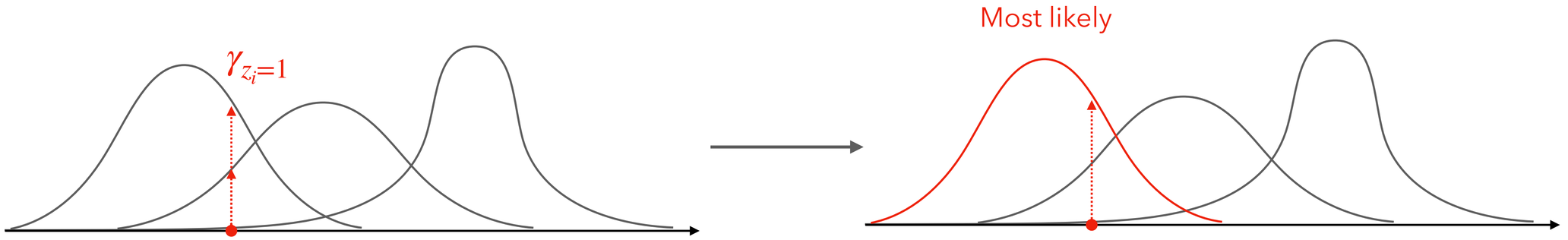
1. Generalized EM

- In the condition of the the Generalized EM, we do not seek the optimal value of θ but only an incremental improvement from the previous step:

$$Q(\theta^{(t)}, \theta^{(t-1)}) > Q(\theta^{(t-1)}, \theta)$$

2. Hard EM / Viterbi Training

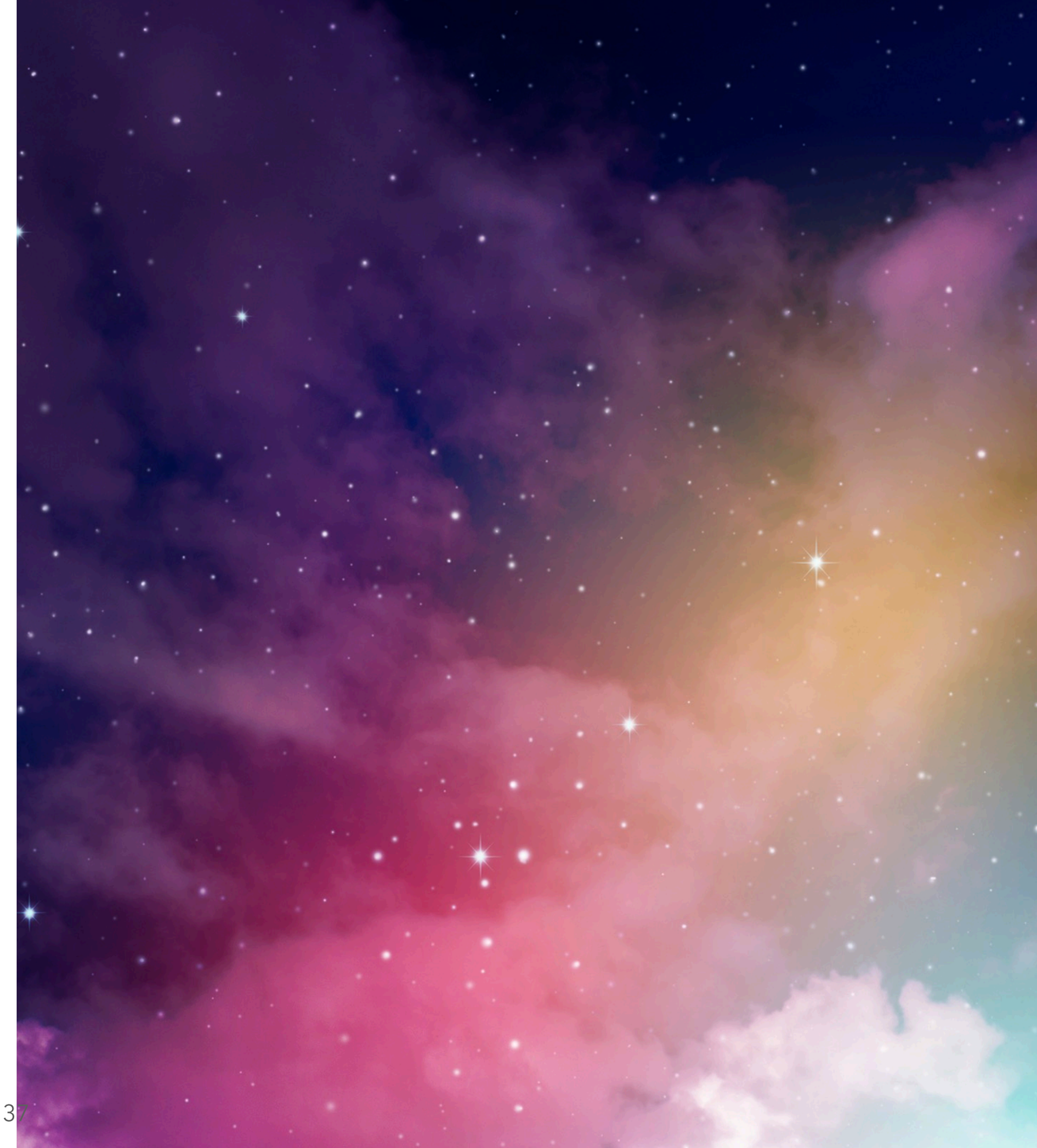
- So far, what we have seen is called the **Soft EM**. In **Hard EM** or **Viterbi Training.** In hard EMs, we make hard decisions for the Z's: $max_{\theta,Z} P(X, Z, \theta)$
- In this case, we do not consider a likelihood weighted over all possible **Z** with their probabilities, but we simply select the most probable Z and move forward.
- The k-Means algorithm is in fact an example of hard EM with $X \sim \mathcal{N}(\mu, I)$, I being the identity covariance matrix



2. Hard EM / Viterbi Training

- Hard EM is easier to implement
- But it does not take into account multiple possibilities for Z, which is a problem if our knowledge of Z is not good enough

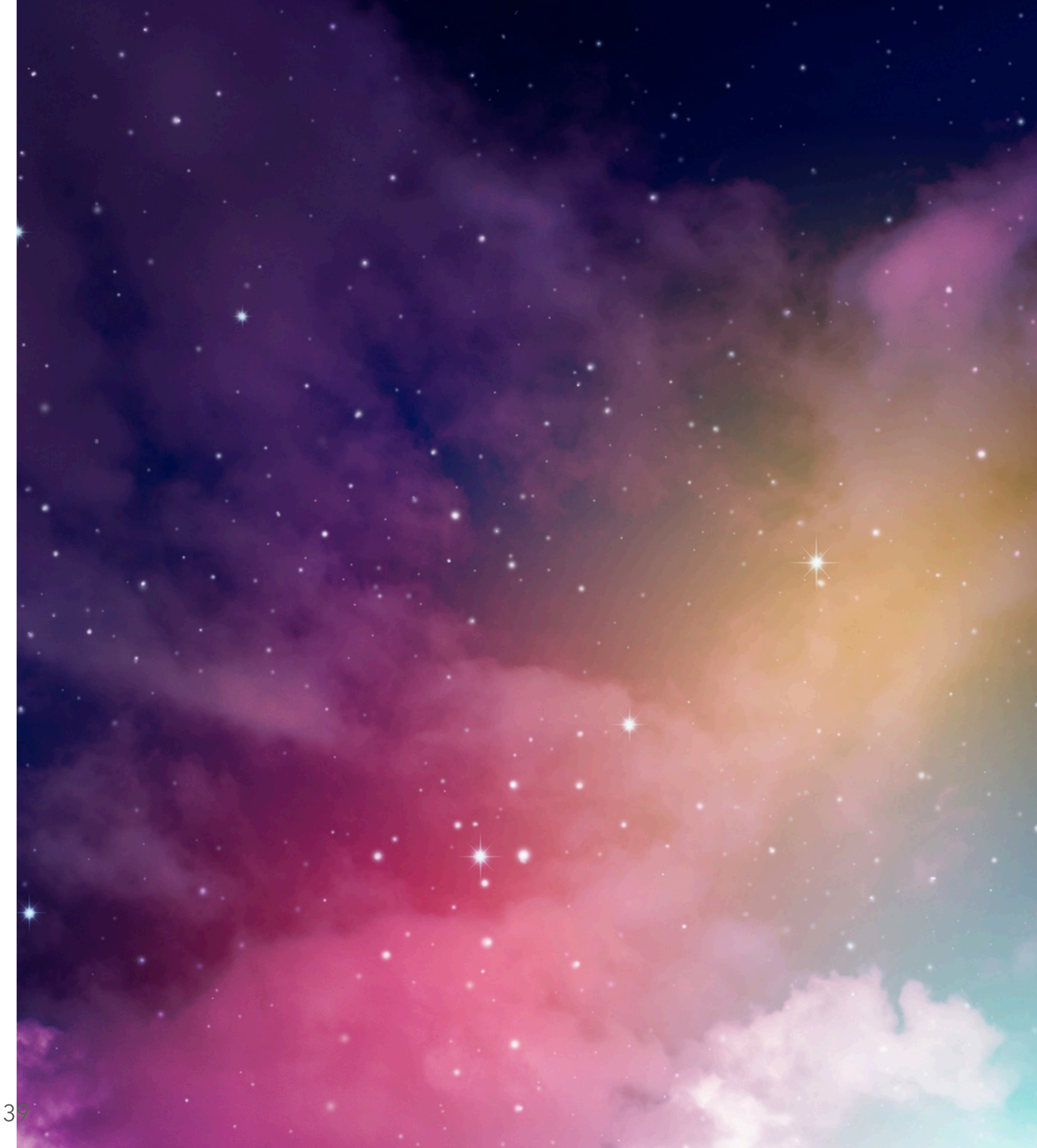
VI.
Limits of EM



VI. Limits of EM

- EM is « initialization-dependent », and converges to local optimum
- EM can be initialized with k-Means parameters:
  - The mean of each cluster identified by k-Means gives μ_k
  - We can compute the within-cluster covariances to identify σ_k
  - We can compute the fraction of data attributed to each cluster to identify w_k
- Highly correlated features might prevent the EM from converging
- It’s not always possible to obtain a full covariance matrix for each gaussian, but diagonal covariance matrix can help

VII.
Applications of EM for GMM



VII. Applications of EM for GMM

- GMMs can be used in unsupervised learning tasks (clustering)

VII. Applications of EM for GMM

- Juan example

VII. Applications of EM for GMM

- GMMs are widely used in speech, for example in gender detection, where one GMM for each gender can be fitted on MFCCs, and we attribute the sample to the GMM with the highest likelihood

VII. Applications of EM for GMM

EM on GMM for gender detection

Recordings

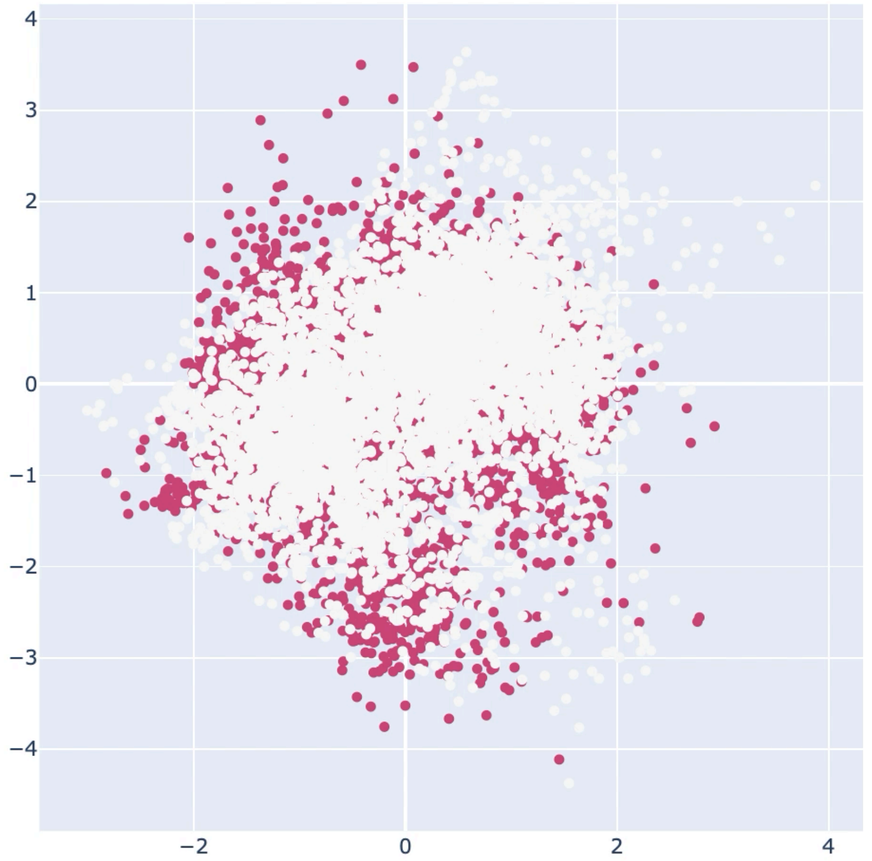
Male recording ▶ 0:00 / 0:10

Female recording ▶ 0:00 / 0:10

Parameters

Number of components: 1 2 3 4 5 6 7 8

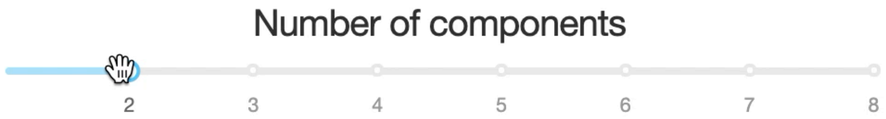
Number of iterations: No EM 10 20 30 40 50



VII. Applications of EM for GMM

- The k-Means, once applied on images or other signals, are called **Vector Quantization** (VQ) and can be used as compression method for images for example, which prevents from storing the value of each pixel, but simply the clusters and the values identified by EM.

VII. Applications of EM for GMM



Original image

Compressed image

VII. Applications of EM for GMM

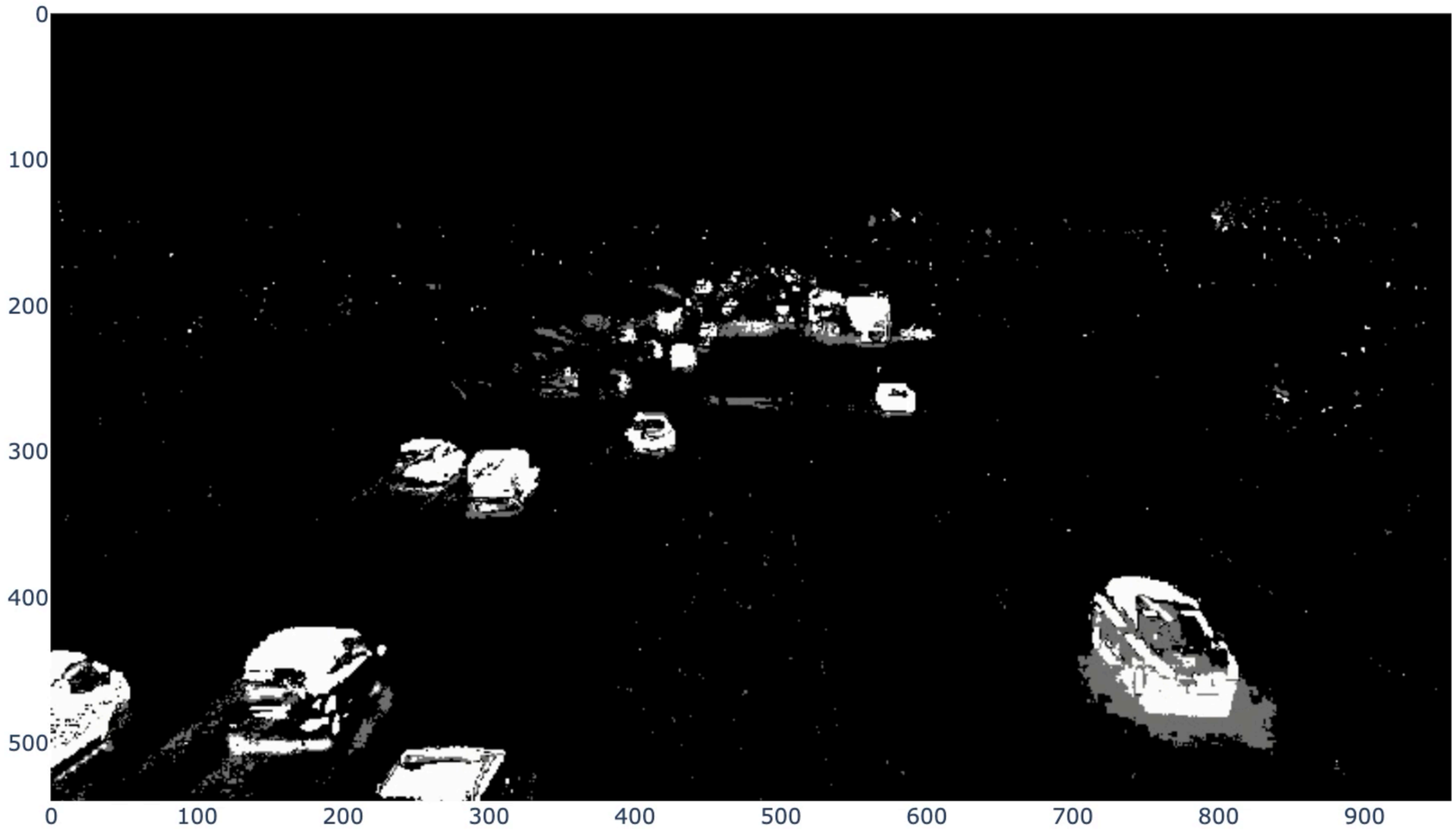
- GMMs are also used for background subtraction in computer vision for example, where the background is a given cluster, and the objects to keep are another cluster.

Figure 3.a

Figure 3.b

As shown in the figure 3.a the pixels corresponding to the road which is background in the image do not undergo state changes, as a result the value 0 (black) is attributed and appears black as shown in figure 3.b. The pixels corresponding to cars undergo drastic changes in state, so the value 1 (white) is attributed and cars appears white as shown in figure 3.b

VII. Applications of EM for GMM



Thank you for your attention
Questions?

VIII. Resources

- MLE of single Gaussian, http://jrmeyer.github.io/machinelearning/2017/08/18/mle.html
- MLE of GMMs, https://stephens999.github.io/fiveMinuteStats/intro_to_em.html
- EM algorithm and variants: an informal tutorial, *Alexis Roche*
- (Hard) Expectation Maximization, *David McAllester*, https://ttic.uchicago.edu/~dmcallester/ttic101-07/lectures/em/em.pdf
- Short Note on EM, *Brendan O'Connor,* https://www.cs.cmu.edu/~tom/10601_fall2012/recitations/em.pdf
- Vector Quantization, *David Forsyth,* http://luthuli.cs.uiuc.edu/~daf/courses/CS-498-DAF-PS/Lecture%2012%20-%20K-means,%20GMMs,%20EM.pdf
- Background subtraction with GMMs, *D. Hari Hara Santosh, P. Venkatesh, P. Poornesh, L. Narayana Rao, N. Arun Kumar,* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.649.8642&rep=rep1&type=pdf