Bayesian Machine Learning

03/05/21 - François HU

Outline

1 **Bayesian statistics**
- define a probabilistic model
- apply bayesian inference
- conjugate priors

2 **Latent variable models**
- define latent variable and apply them to simplify probabilistic model
- cluster data with latent models like GMM
- train probabilistic models with EM-algorithm

3 **Variational Inference**
- apply variational inference for probabilistic models
- understand variational interpretation of LDA
- application of LDA to text mining

4 **Markov Chain Monte Carlo**
- train / do inference almost any probabilistic model with MCMC
- pros and cons of MCMC / VI

5 **Extensions and oral presentations**

0 Evaluation & conjugate prior

Evaluation

Group project

- The evaluation will consist of a **group project** based on a research article : one article for each group (composed of 2 persons)

- During the last lecture, each group will send me the **codes** and give an **oral presentation** in front of the class. Even if the article is mostly theoretical, each presentation should be understandable by other groups (The clarity of the speech will be analysed).

- Initiatives like **more experimentations** or identifying the limits of the article will be greatly appreciated. You are welcome to consult other research articles (it should be cited at the end of your presentation) to boost your knowledge (but don't forget that the proposed paper is the core of your presentation)

- The **evaluation** is as follows :

    - **40% on the clarity of the code** (example : many comments, along with understandable variables/functions names. You can use Jupyter Notebook which might have the advantage to be easy to read for the users). When I run your code, it should be easy to run and easy to understand :)

    - **60% on the clarity of the oral presentation**. Less maths but more experimentations and intuitions. At the beginning a big introduction is expected in order to be understandable by other groups.

Evaluation
Research articles

1. « **Algorithmic Assurance : An Active Approach to Algorithmic Testing using Bayesian Optimisation** » **(NeurIPS 2018)**
   Shivapratap Gopakumar, Sunil Gupta, Santu Rana, Vu Nguyen and Svetha Svekatesh. 2018
   key notions : Bayesian optimisation, algorithmic testing

2. « **A Bayesian Nonparametric Approach for Multi-label Classification** » **(JMLR 2016)**
   Vu Nguyen, Sunil Gupta, Santu Rana, Cheng Li and Svetha Venkatesh. 2016
   key notions : Nonparametric Bayesian, multi-label classification

3. « **What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision ?** » **(NeurIPS 2017)**
   Alex Kendall and Yarin Gal. 2017
   key notions : Uncertainties, Bayesian Deep Learning

4. « **Deep Bayesian Active Learning with Image Data** » **(ICML 2017)**
   Yarin Gal, Riashat Islam and Zoubin Ghahramani. 2017
   key notions : Active learning, Bayesian Deep Learning

Conjugate priors : notebook

website : https://curiousml.github.io/

// EPITA - École pour l'informatique et les techniques avancées (2020 - …)

- Master of Science in Artificial Intelligence Systems : **Bayesian Machine Learning** by François HU
    - **Training session / prerequisite :** [Statistics with python], [Data]
    - **Lecture 1** : [Bayesian statistics]
    - **Practical work 1** : [Conjugate distributions] [Correction]
    - **Lecture 2** : (soon available)
    - **Practical work 2** : (soon available)
    - **Lecture 3** : (soon available)
    - **Practical work 3** : (soon available)
    - **Lecture 4** : (soon available)
    - **Practical work 4** : (soon available)
    - **Lecture 5** : (soon available)

TODO

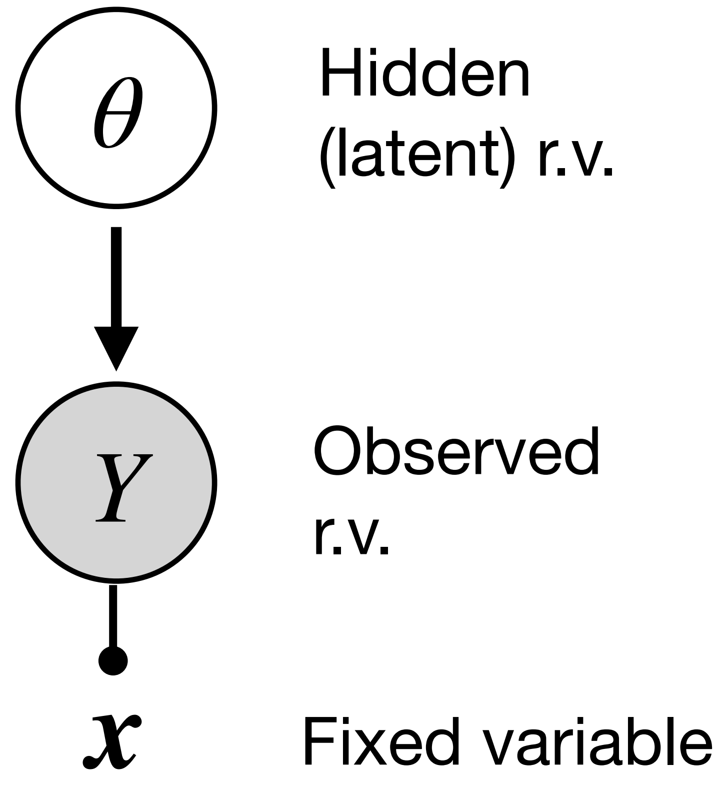
1 Latent variable models and mixture models

1. Latent Variable Models

Spoilers

Latent variable models : a statistical model that links a set of **observable** variables to a set of **unobservable (latent)** variables

Example : Bayesian Linear regression



Other latent variable models : unsupervised methods

- **Clustering** models
- **Dimensionality reduction** models

Questions :

- **Why do we need latent variable models ? simpler models (so fewer parameters) without reducing its flexibility**
- **How to train these models ? next section**
- **Any limitations of the proposed training method ? last section**

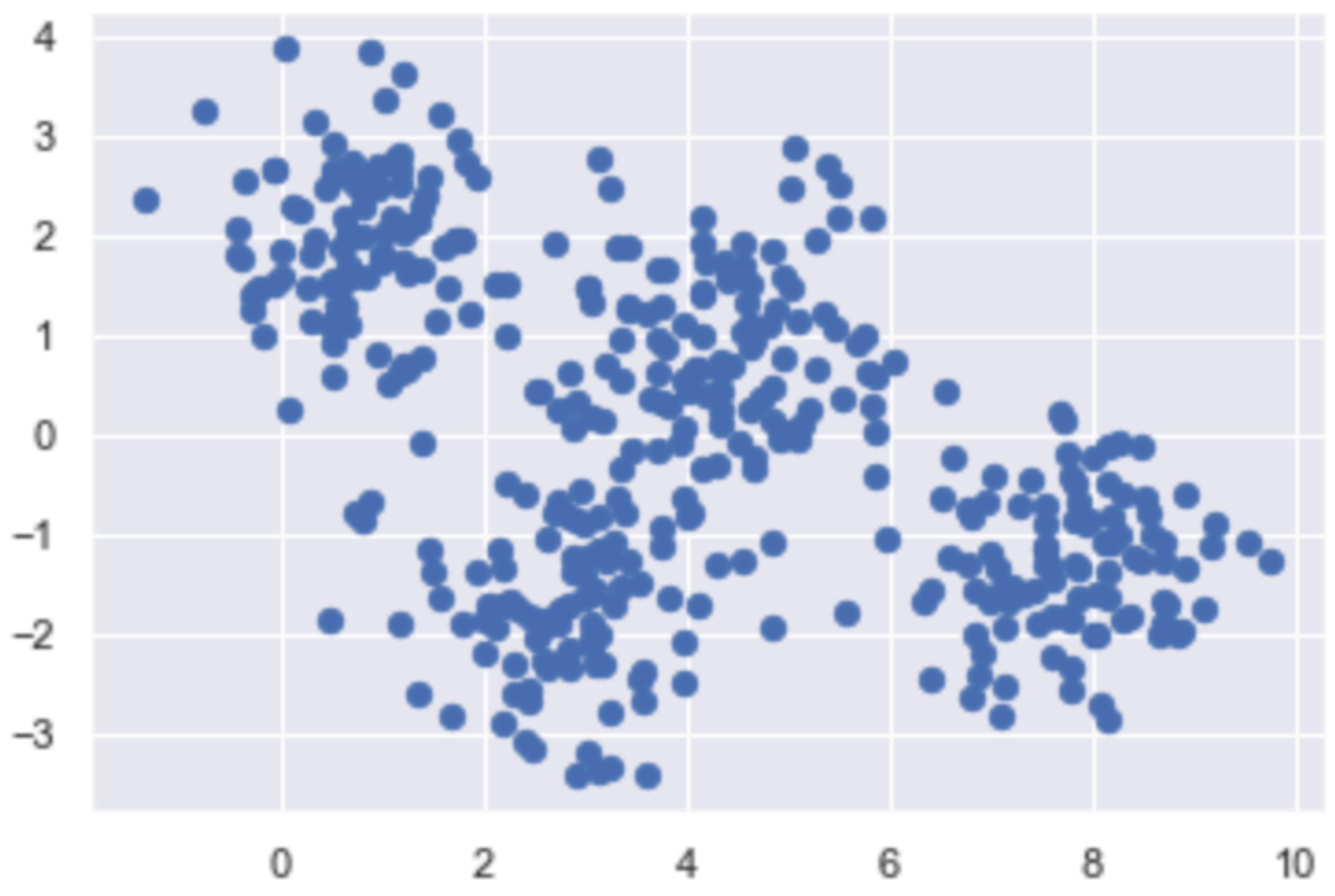
1. Latent Variable Models

Mixture models : Definition

Mixture models : a probabilistic model representing a **linear combination** of different distributions

Mixture modeling provides the freedom / flexibility to model the unknown pdf. Downside : more parameters

Example : synthetic data



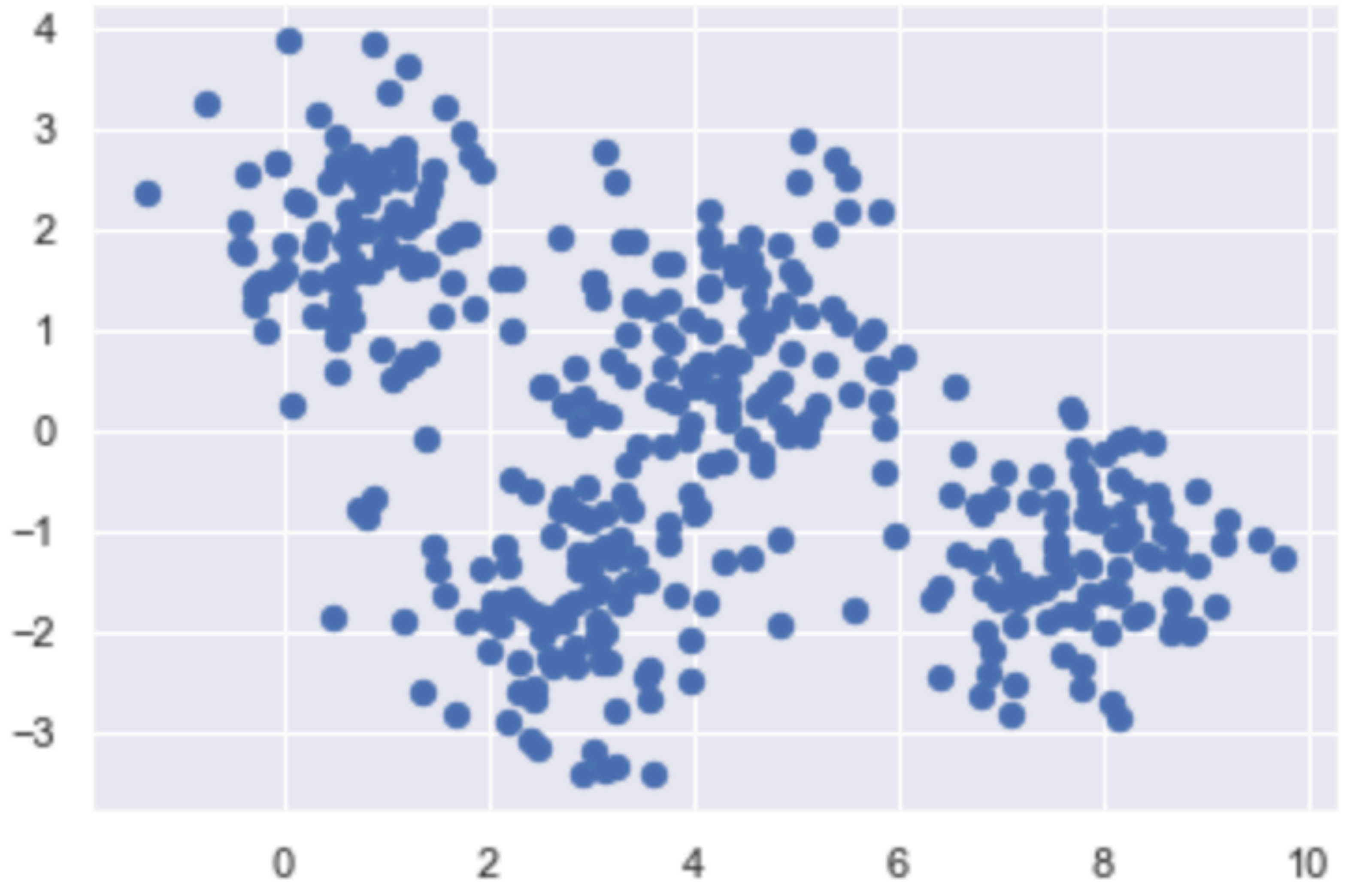
1. Latent Variable Models

Mixture models : Definition

Mixture models : a probabilistic model representing a **linear combination** of different distributions

Mixture modeling provides the freedom / flexibility to model the unknown pdf. Downside : more parameters

Example : synthetic data



Let's fit a gaussian !

$$\mathcal{N}(\mu, \Sigma)$$

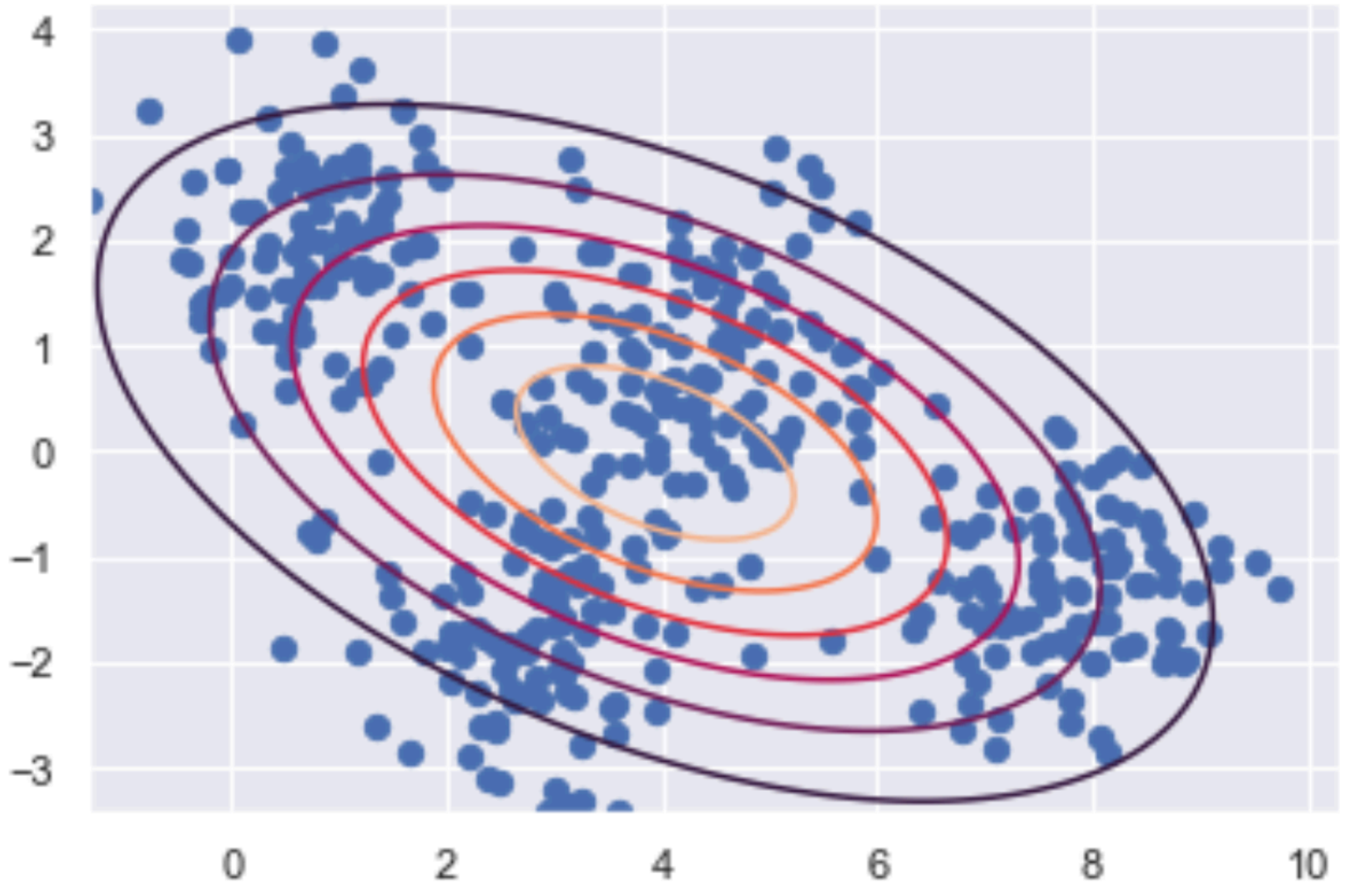
1. Latent Variable Models

Mixture models : Definition

Mixture models : a probabilistic model representing a **linear combination** of different distributions

Mixture modeling provides the freedom / flexibility to model the unknown pdf. Downside : more parameters

Example : synthetic data



Let's fit a gaussian !

$$\mathcal{N}(\mu, \Sigma)$$

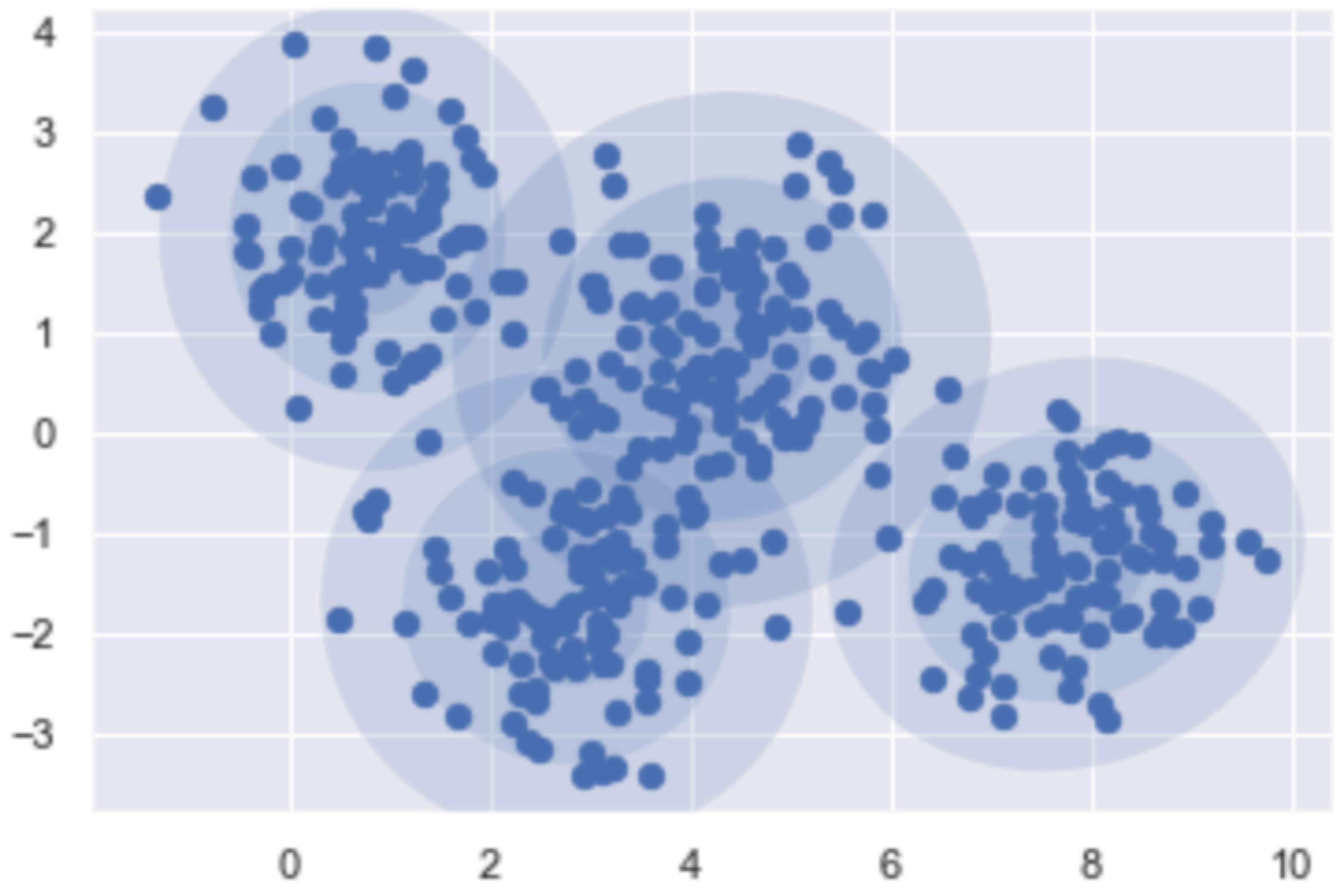
1. Latent Variable Models

Gaussian Mixture Model : Definition

Mixture models : a probabilistic model representing a **linear combination** of different distributions

Mixture modeling provides the freedom / flexibility to model the unknown pdf. Downside : more parameters

Example : synthetic data



Let's fit a gaussian !

$$\mathcal{N}(\mu, \Sigma)$$

We want to fit a Gaussian Mixture Model (GMM) !

$$\sum_{k=1}^{4} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k)$$

parameters : $\{\pi_k, \mu_k, \Sigma_k\}_{k \in \{1,\ldots,4\}} =: \theta$

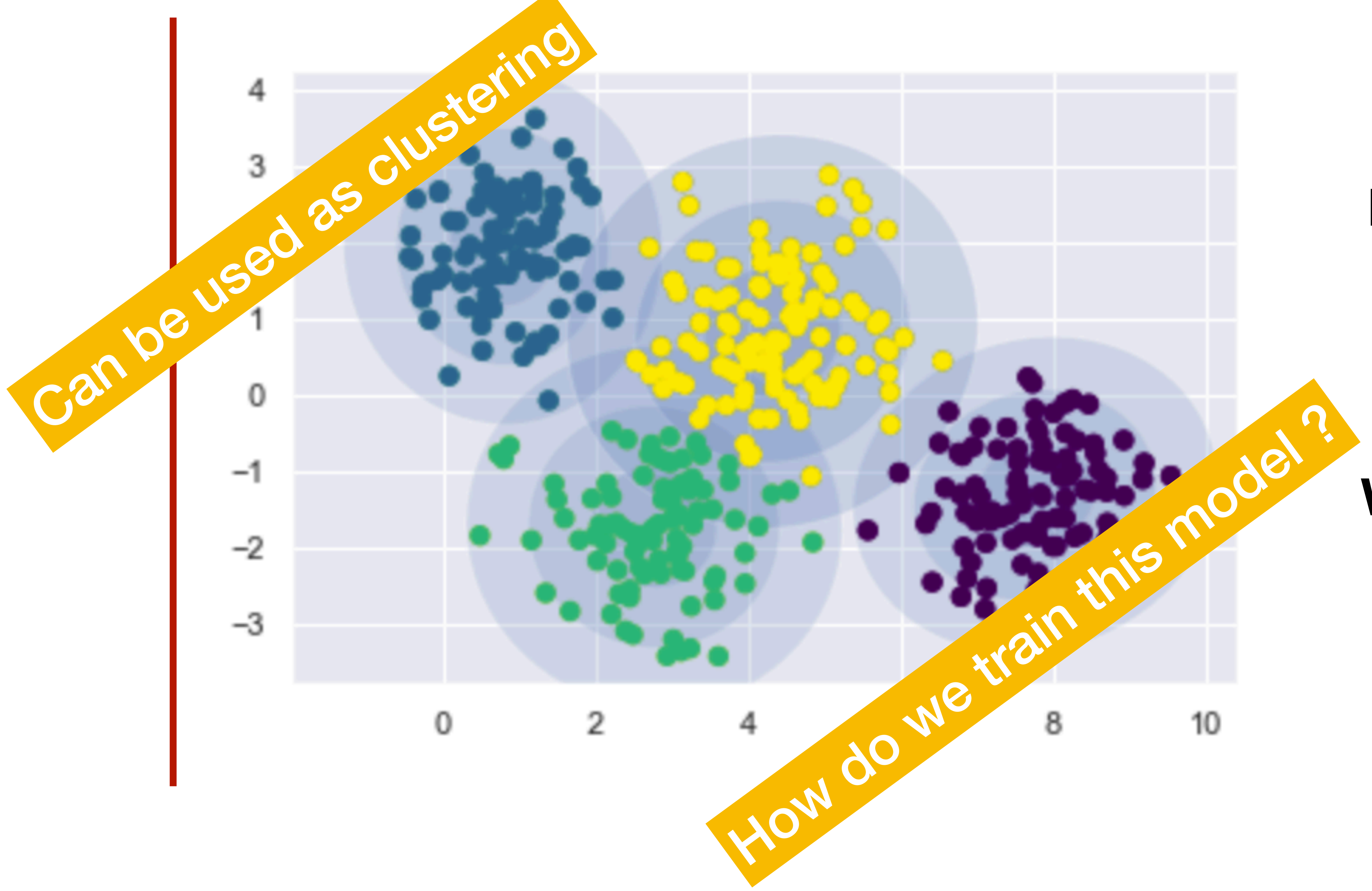
1. Latent Variable Models

Gaussian Mixture Model : Definition

Mixture models : a probabilistic model representing a **linear combination** of different distributions

Mixture modeling provides the freedom / flexibility to model the unknown pdf. Downside : more parameters

Example : synthetic data



Let's fit a gaussian !

$$\mathcal{N}(\mu, \Sigma)$$

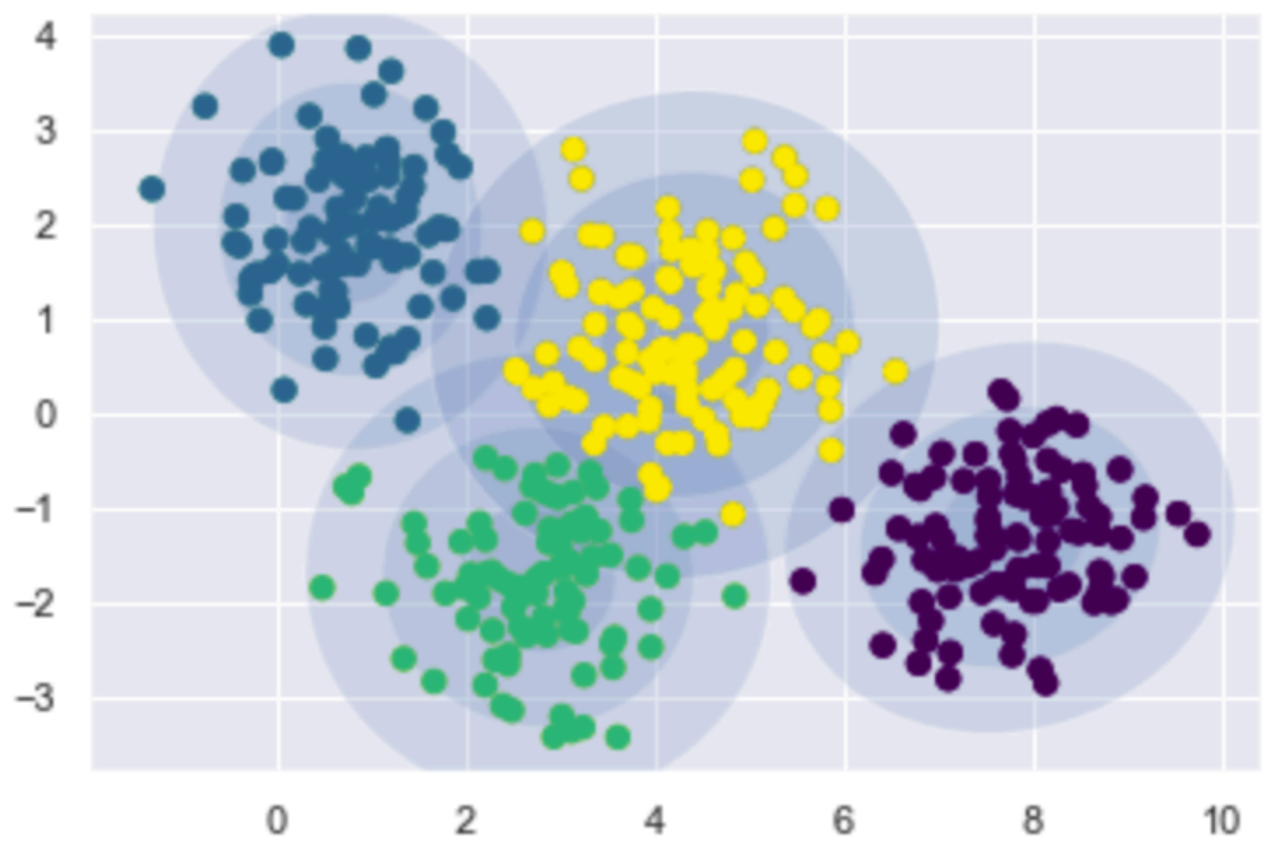
We want to fit a Gaussian Mixture Model (GMM) !

$$\sum_{k=1}^{4} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k)$$

parameters : $\{\pi_k, \mu_k, \Sigma_k\}_{k \in \{1,\ldots,4\}} =: \theta$

1. Latent Variable Models

Gaussian Mixture Model : Training with MLE ?

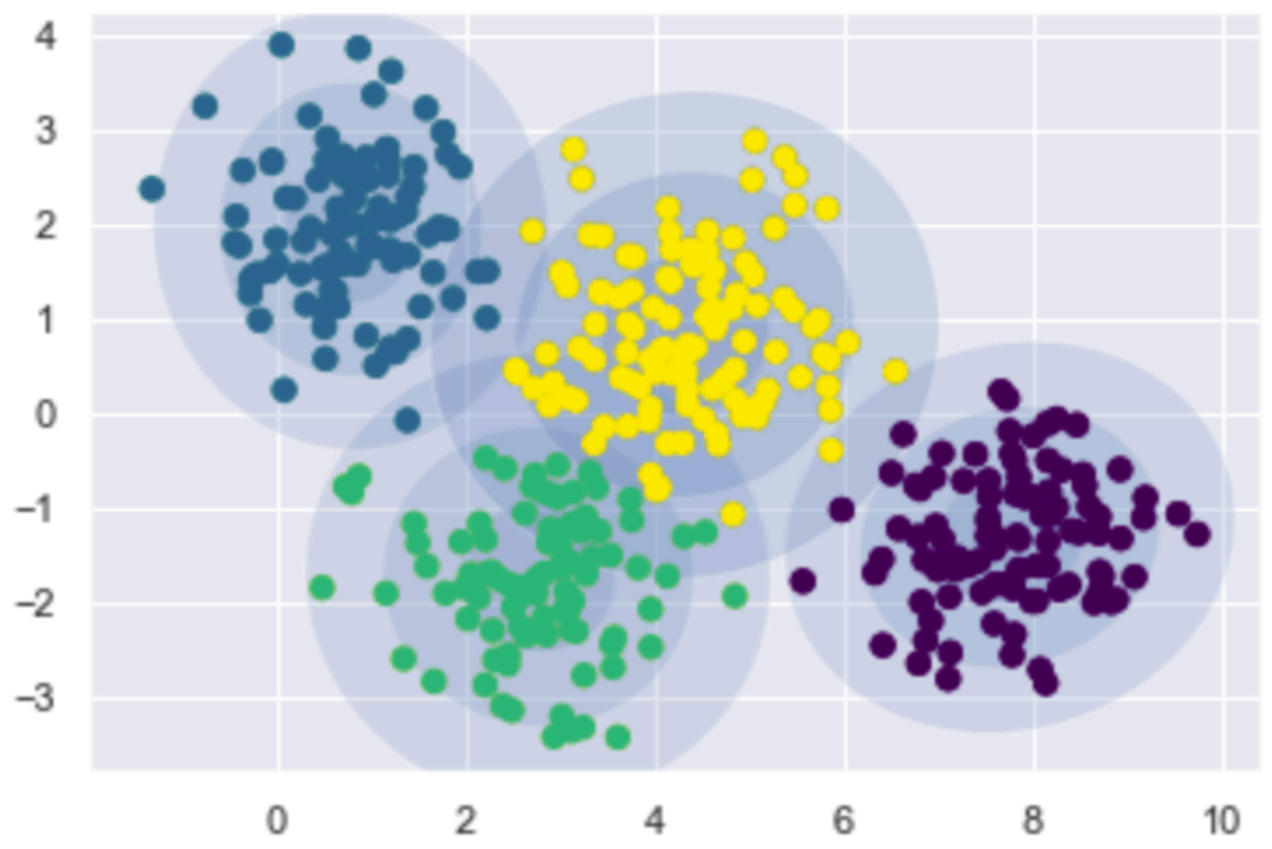


We want to fit a Gaussian Mixture Model !

$$\sum_{k=1}^{4} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k) \qquad \text{parameters : } \{\pi_k, \mu_k, \Sigma_k\} =: \theta$$

1. Latent Variable Models

Gaussian Mixture Model : Training with MLE ?



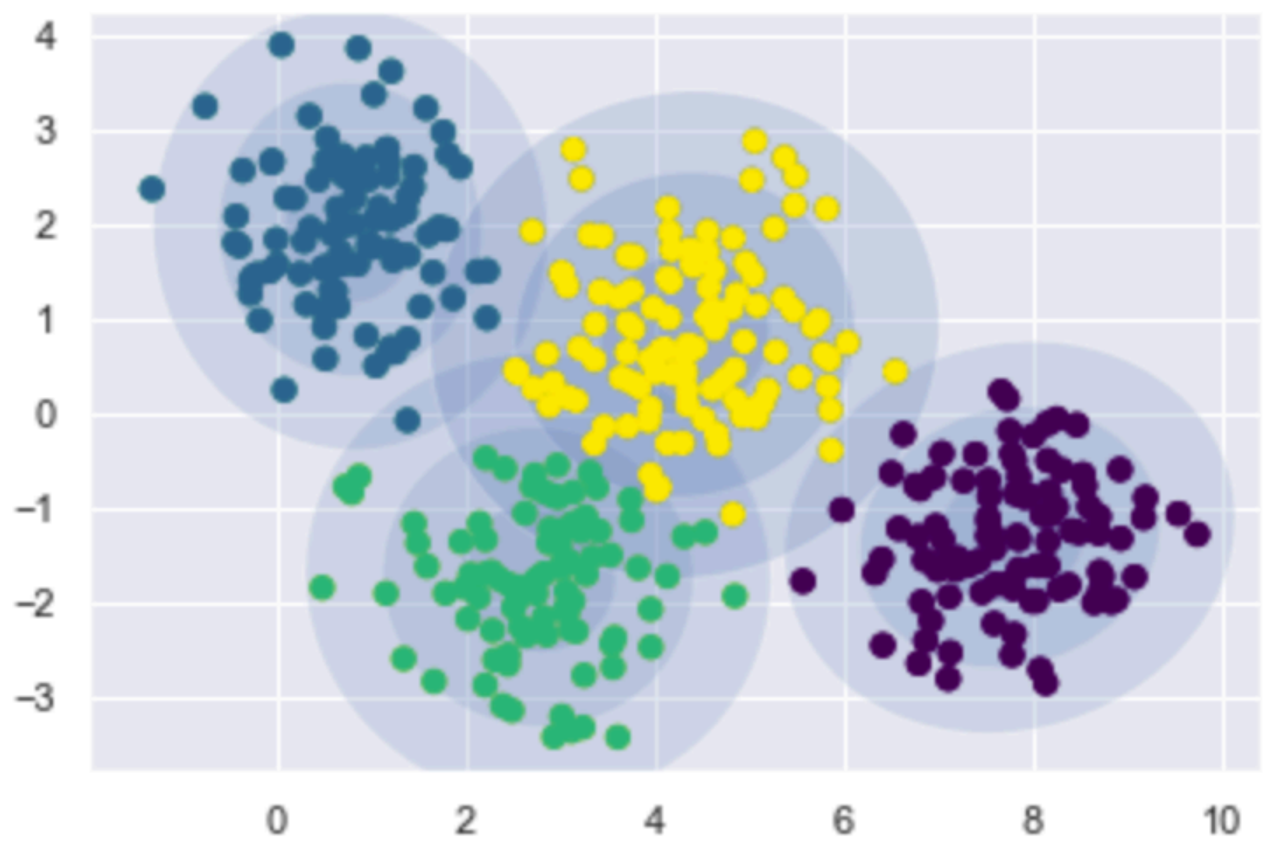
We want to fit a Gaussian Mixture Model !

$$\sum_{k=1}^{4} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k) \qquad \text{parameters : } \{\pi_k, \mu_k, \Sigma_k\} =: \theta$$

We can train our model by Maximum Likelihood Estimation (MLE) + Stochastic Gradient Descent (SGD) :

1. Latent Variable Models

Gaussian Mixture Model : Training with MLE ?



We want to fit a Gaussian Mixture Model !

$$\sum_{k=1}^{4} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k) \qquad \text{parameters : } \{\pi_k, \mu_k, \Sigma_k\} =: \theta$$

We can train our model by Maximum Likelihood Estimation (MLE) + Stochastic Gradient Descent (SGD) :

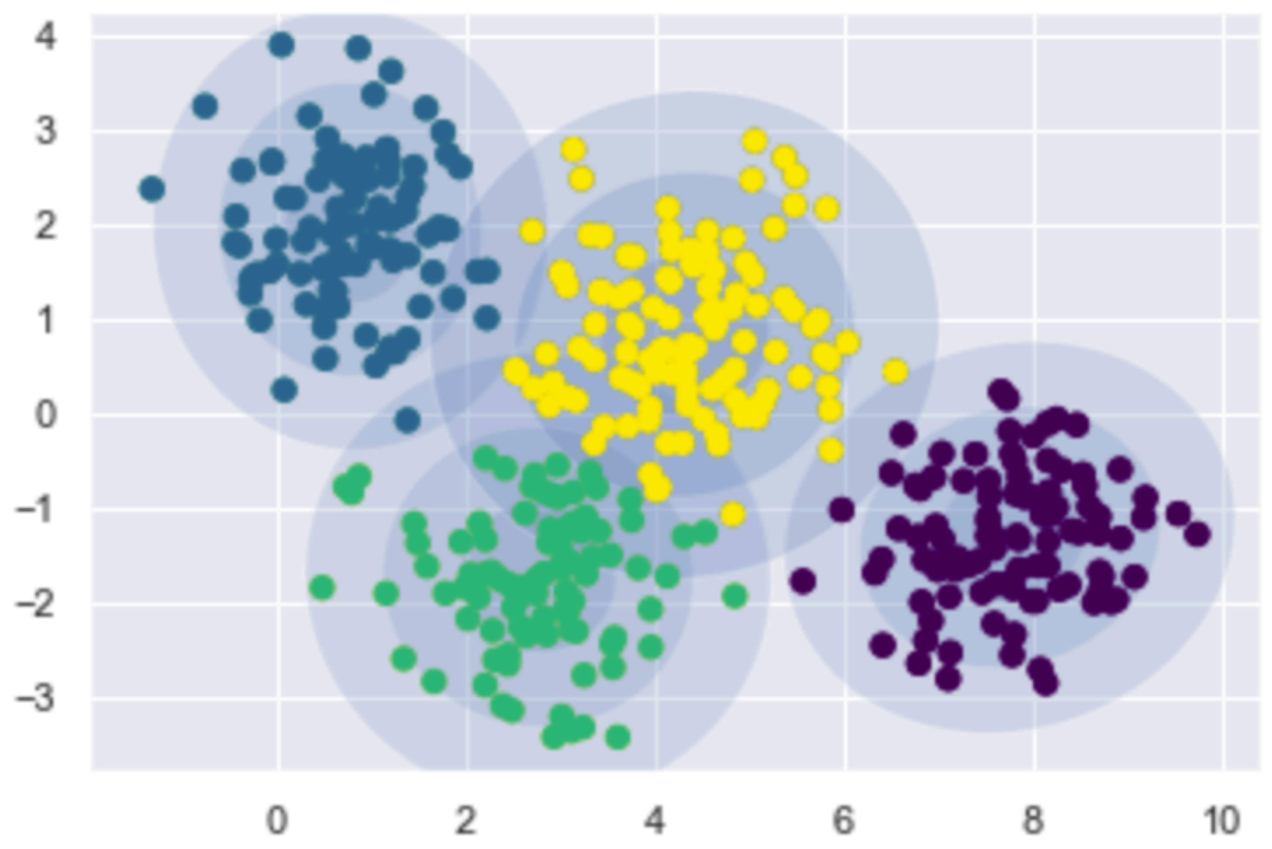
$$\max_\theta \prod_{i=1}^{n} p(x_i | \theta) = \max_\theta \prod_{i=1}^{n} (\pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2) + \pi_3 \mathcal{N}(\mu_3, \Sigma_3) + \pi_4 \mathcal{N}(\mu_4, \Sigma_4))$$

subject to $\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$ with each $\pi_k \geq 0$

with each $\Sigma_k \succ 0$

1. Latent Variable Models

Gaussian Mixture Model : Training with MLE ?



We want to fit a Gaussian Mixture Model !

$$\sum_{k=1}^{4} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k) \qquad \text{parameters : } \{\pi_k, \mu_k, \Sigma_k\} =: \theta$$

We can train our model by Maximum Likelihood Estimation (MLE) + Stochastic Gradient Descent (SGD) :

$$\max_{\theta} \prod_{i=1}^{n} p(x_i \mid \theta) = \max_{\theta} \prod_{i=1}^{n} (\pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2) + \pi_3 \mathcal{N}(\mu_3, \Sigma_3) + \pi_4 \mathcal{N}(\mu_4, \Sigma_4))$$

subject to $\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$ with each $\pi_k \geq 0$

with each $\Sigma_k \succ 0$

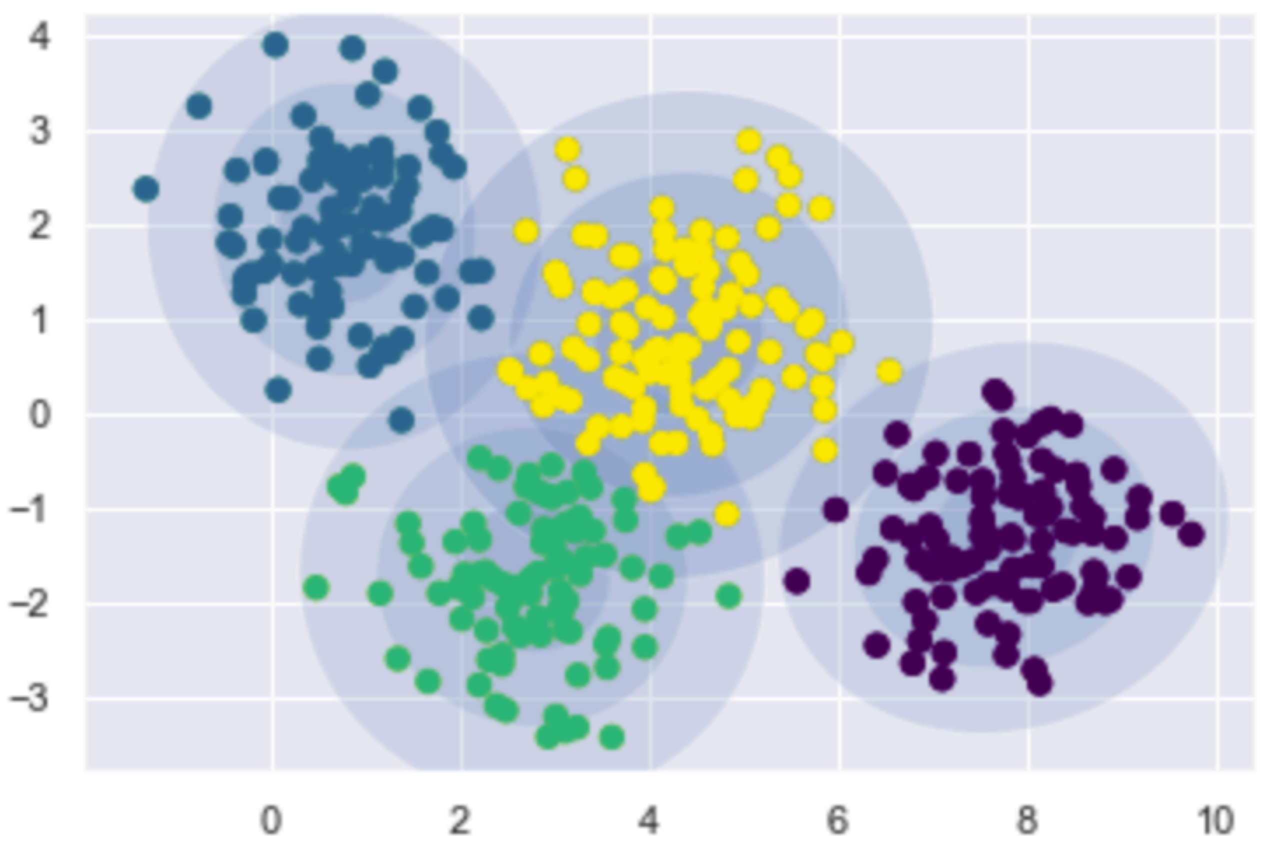
Problem : it turns out that it is too difficult for SGD (cannot maintain this constraint)

Solution : let Σ_k be diagonal

Limit : It can be done but not optimal

1. Latent Variable Models

Gaussian Mixture Model : Training with MLE ?



We want to fit a Gaussian Mixture Model !

$$\sum_{k=1}^{4} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k) \qquad \text{parameters : } \{\pi_k, \mu_k, \Sigma_k\} =: \theta$$

We can train our model by Maximum Likelihood Estimation (MLE) + Stochastic Gradient Descent (SGD) :

$$\max_\theta \prod_{i=1}^{n} p(x_i | \theta) = \max_\theta \prod_{i=1}^{n} (\pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2) + \pi_3 \mathcal{N}(\mu_3, \Sigma_3) + \pi_4 \mathcal{N}(\mu_4, \Sigma_4))$$

subject to $\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$ with each $\pi_k \geq 0$

with each $\Sigma_k \succ 0$

Problem : it turns out that it is too difficult for SGD (cannot maintain this constraint)

Solution : let Σ_k be diagonal

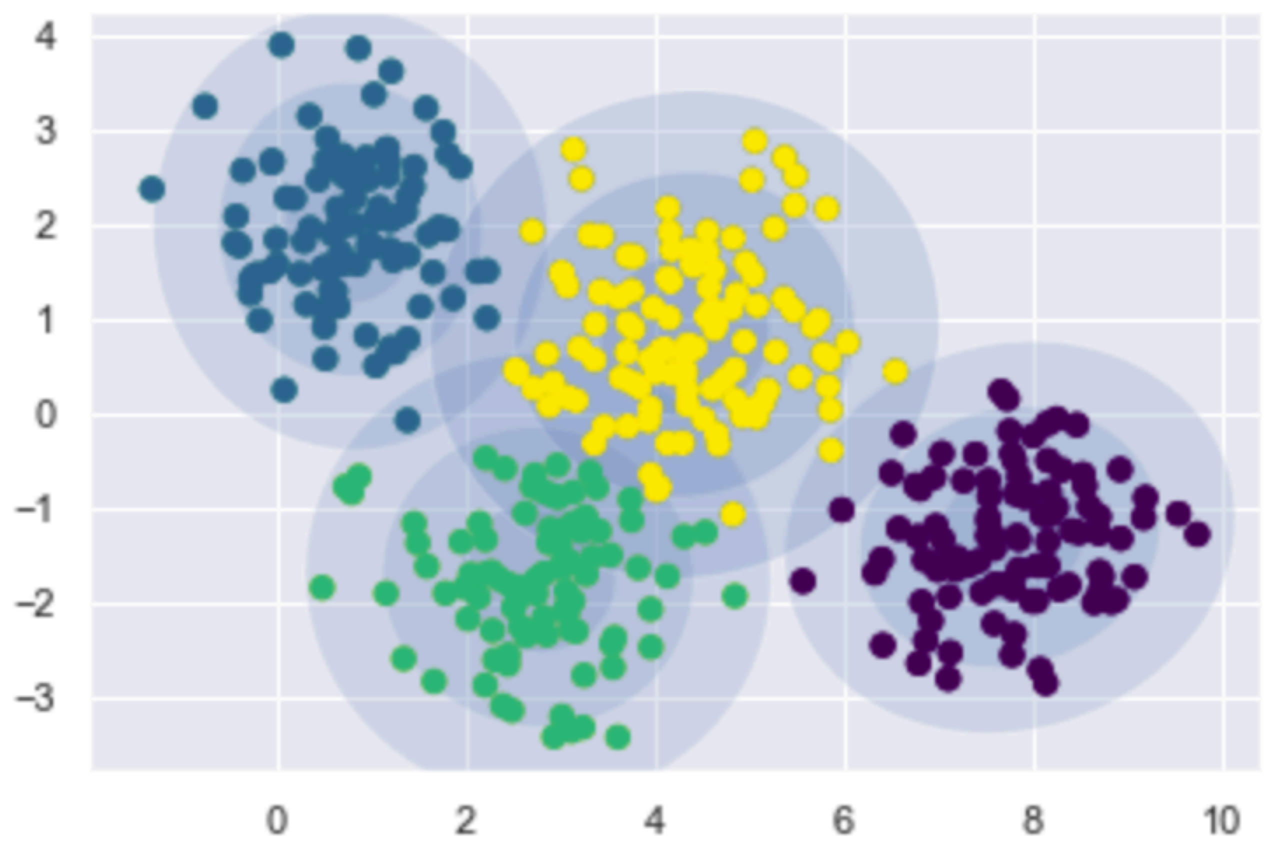
Limit : It can be done but not optimal

EM algorithm

2 Probabilistic clustering and EM-algorithm

2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model



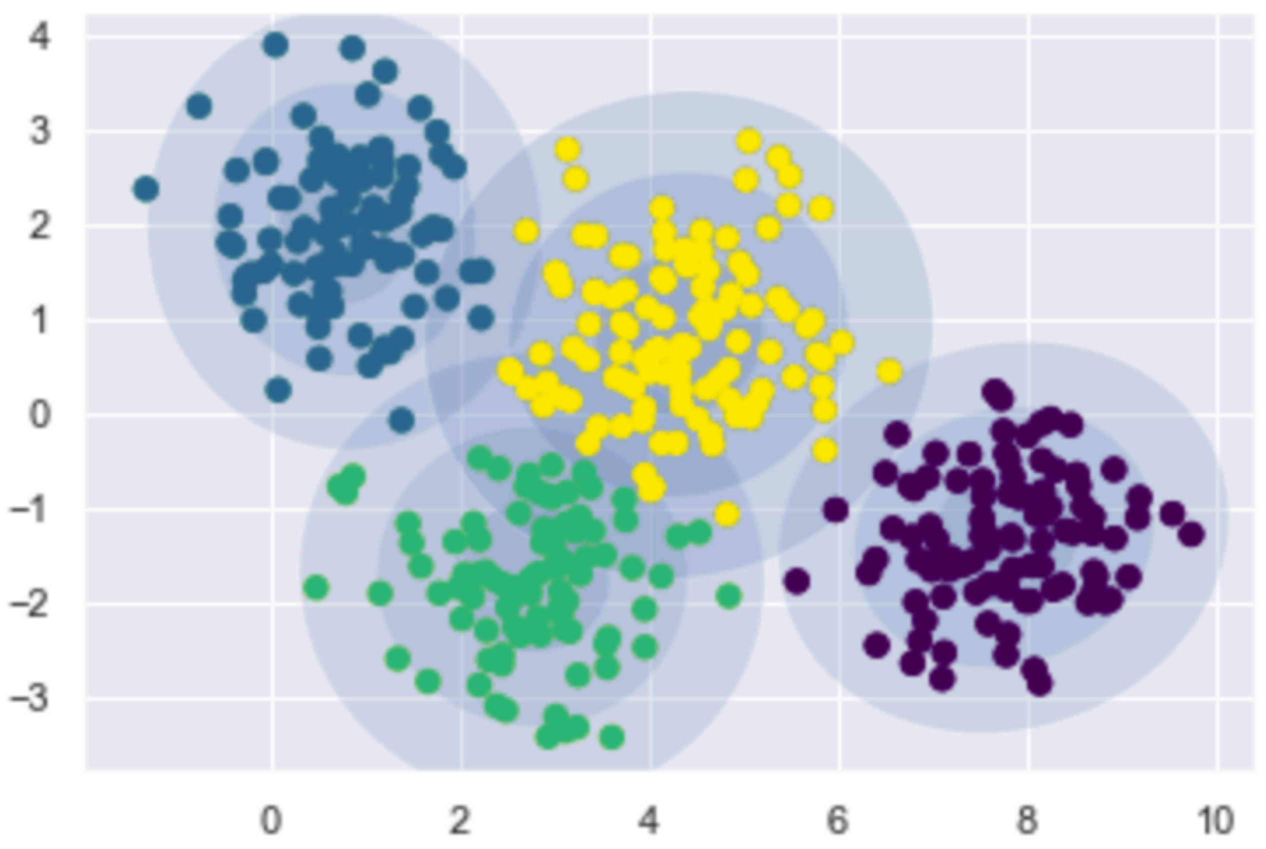
We want to fit a Gaussian Mixture Model !

$$\pi_1\mathcal{N}(\mu_1,\Sigma_1)+\pi_2\mathcal{N}(\mu_2,\Sigma_2)+\pi_3\mathcal{N}(\mu_3,\Sigma_3)+\pi_4\mathcal{N}(\mu_4,\Sigma_4)$$

parameters : $\theta = \{\pi_1,\mu_1,\Sigma_1,\pi_2,\mu_2,\Sigma_2,\pi_3,\mu_3,\Sigma_3,\pi_4,\mu_4,\Sigma_4\}$

2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model



We want to fit a Gaussian Mixture Model !

$$\pi_1\mathcal{N}(\mu_1,\Sigma_1) + \pi_2\mathcal{N}(\mu_2,\Sigma_2) + \pi_3\mathcal{N}(\mu_3,\Sigma_3) + \pi_4\mathcal{N}(\mu_4,\Sigma_4)$$

parameters : $\theta = \{\pi_1,\mu_1,\Sigma_1,\pi_2,\mu_2,\Sigma_2,\pi_3,\mu_3,\Sigma_3,\pi_4,\mu_4,\Sigma_4\}$

Latent variable model for GMM :

$$t \longrightarrow x$$

the source :
from **which** gaussian
{1, 2, 3, 4}
this data came from ?

Gaussian
distribution

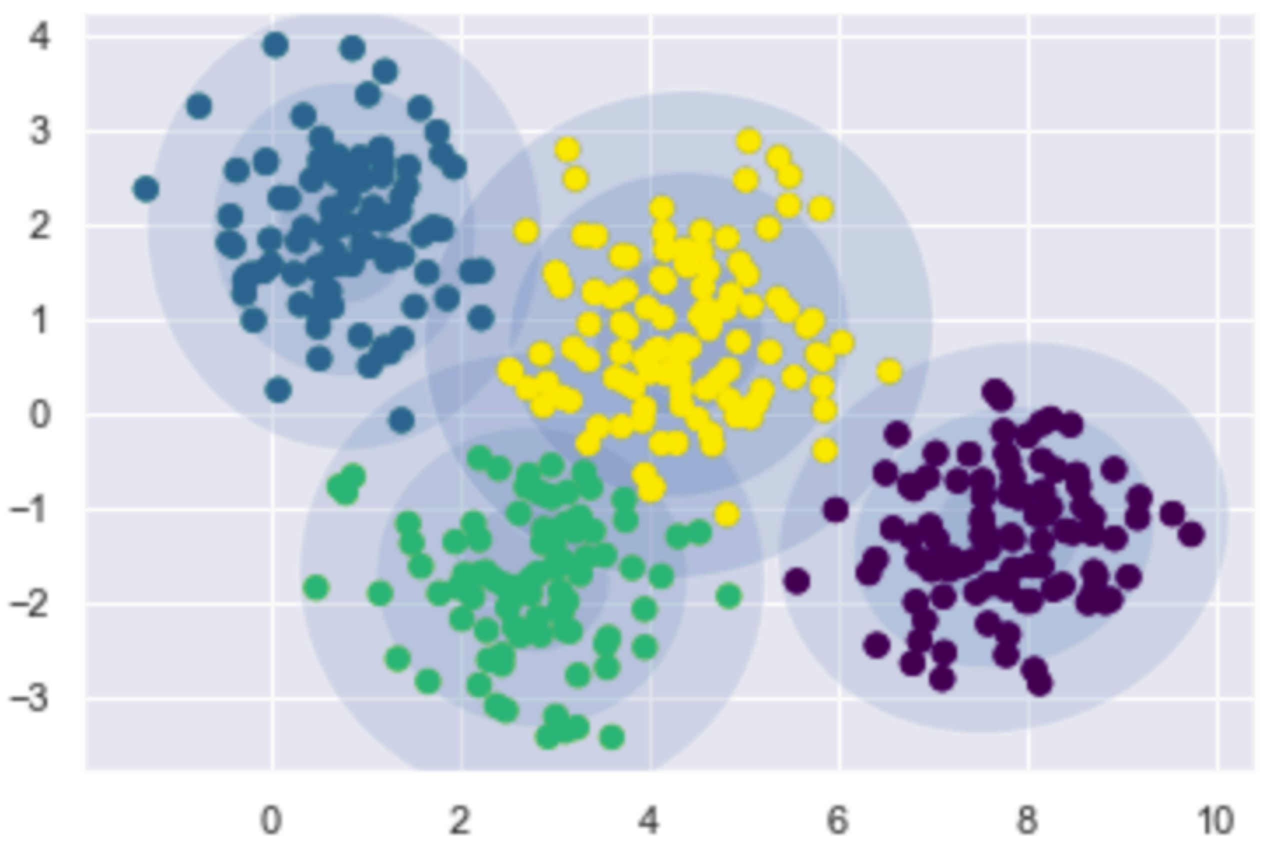
$p(t = k \mid \theta) =$

$p(x \mid t = k, \theta) =$

$p(x \mid \theta) =$

2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model



We want to fit a Gaussian Mixture Model !

$$\pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2) + \pi_3 \mathcal{N}(\mu_3, \Sigma_3) + \pi_4 \mathcal{N}(\mu_4, \Sigma_4)$$

parameters : $\theta = \{\pi_1, \mu_1, \Sigma_1, \pi_2, \mu_2, \Sigma_2, \pi_3, \mu_3, \Sigma_3, \pi_4, \mu_4, \Sigma_4\}$

Latent variable model for GMM :

the source :
from **which** gaussian
{1, 2, 3, 4}
this data came from ?

Gaussian
distribution

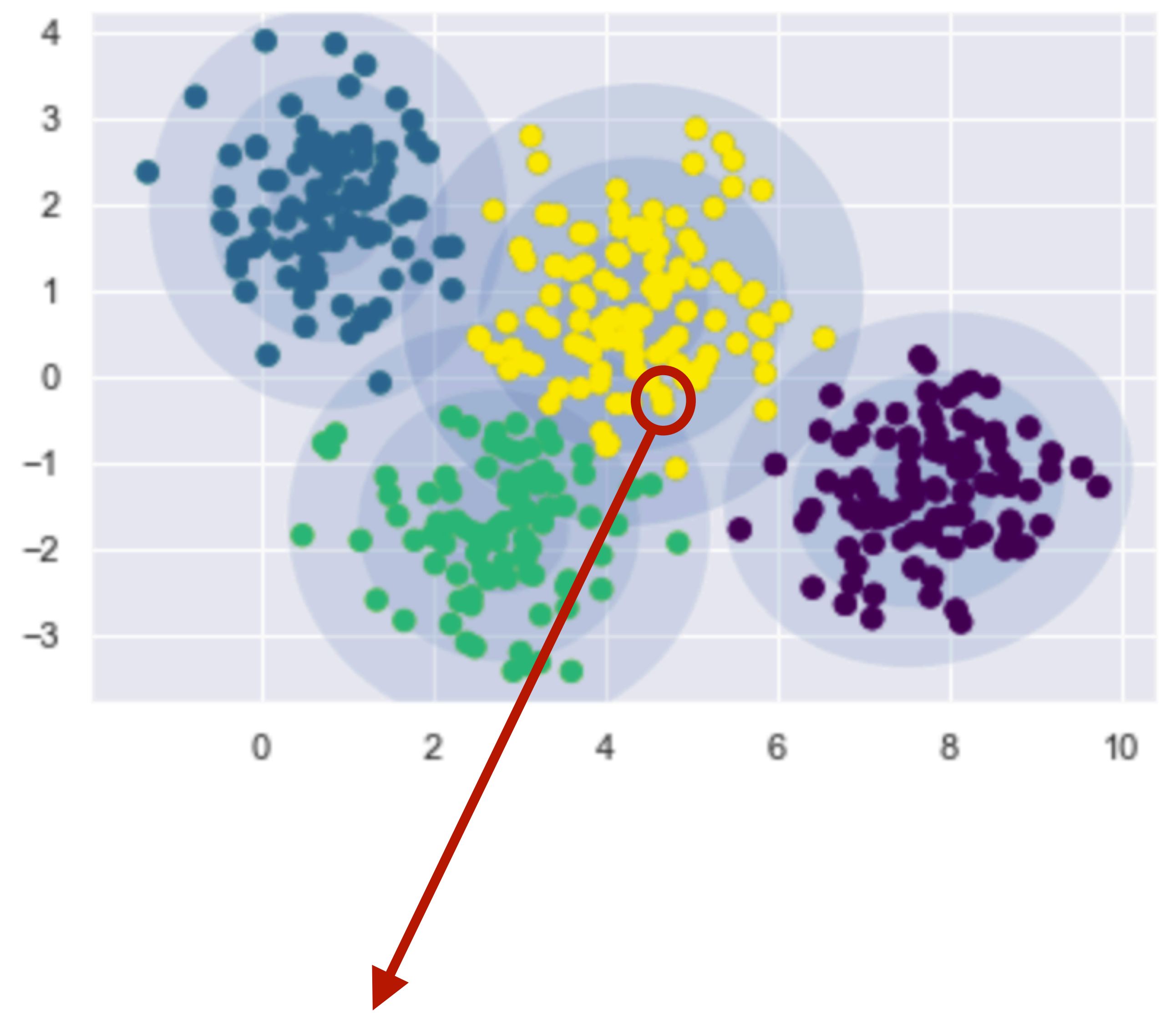
$$p(t = k \,|\, \theta) = \pi_k$$

$$p(x \,|\, t = k, \theta) = \mathcal{N}(x \,|\, \mu_k, \Sigma_k)$$

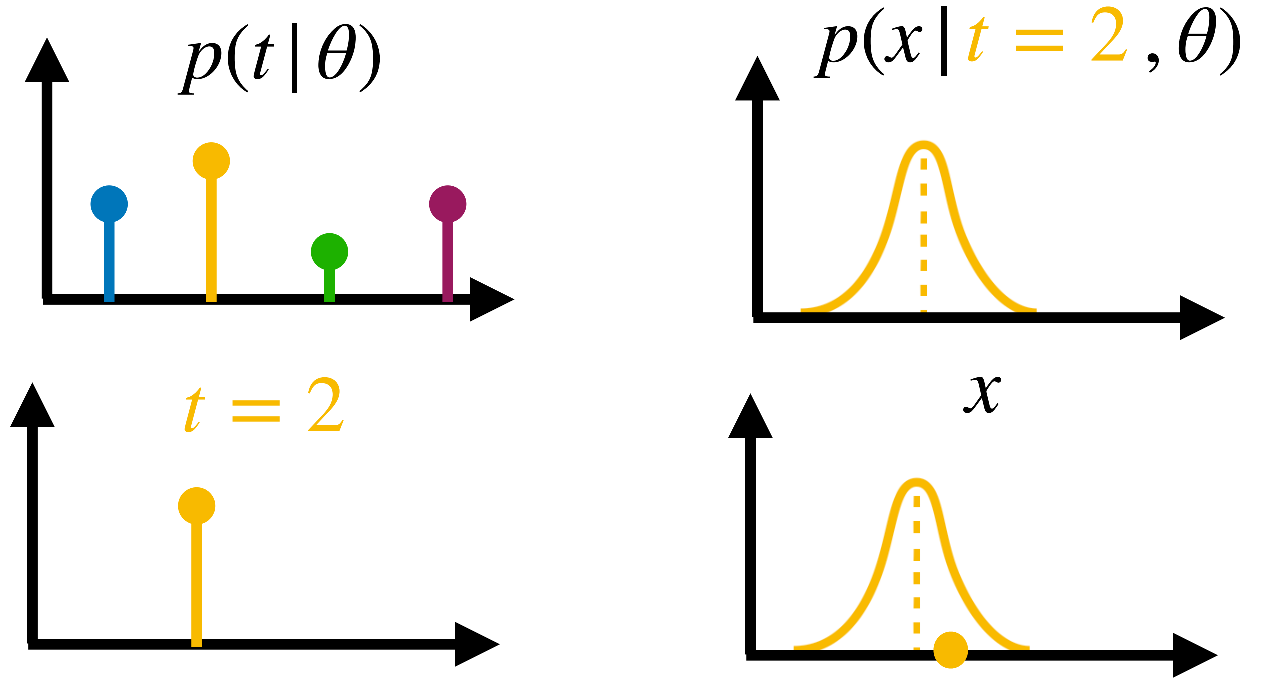
$$p(x \,|\, \theta) = \sum_{k=1}^{4} p(x \,|\, t = k, \theta) p(t = k \,|\, \theta)$$

2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model



We assume that this x is generated as follows :



We want to fit a Gaussian Mixture Model !

$$\pi_1\mathcal{N}(\mu_1,\Sigma_1)+\pi_2\mathcal{N}(\mu_2,\Sigma_2)+\pi_3\mathcal{N}(\mu_3,\Sigma_3)+\pi_4\mathcal{N}(\mu_4,\Sigma_4)$$

parameters : $\theta=\{\pi_1,\mu_1,\Sigma_1,\pi_2,\mu_2,\Sigma_2,\pi_3,\mu_3,\Sigma_3,\pi_4,\mu_4,\Sigma_4\}$

Latent variable model for GMM :

the source :
from **which** gaussian
{1, 2, 3, 4}
this data came from ?

Gaussian
distribution

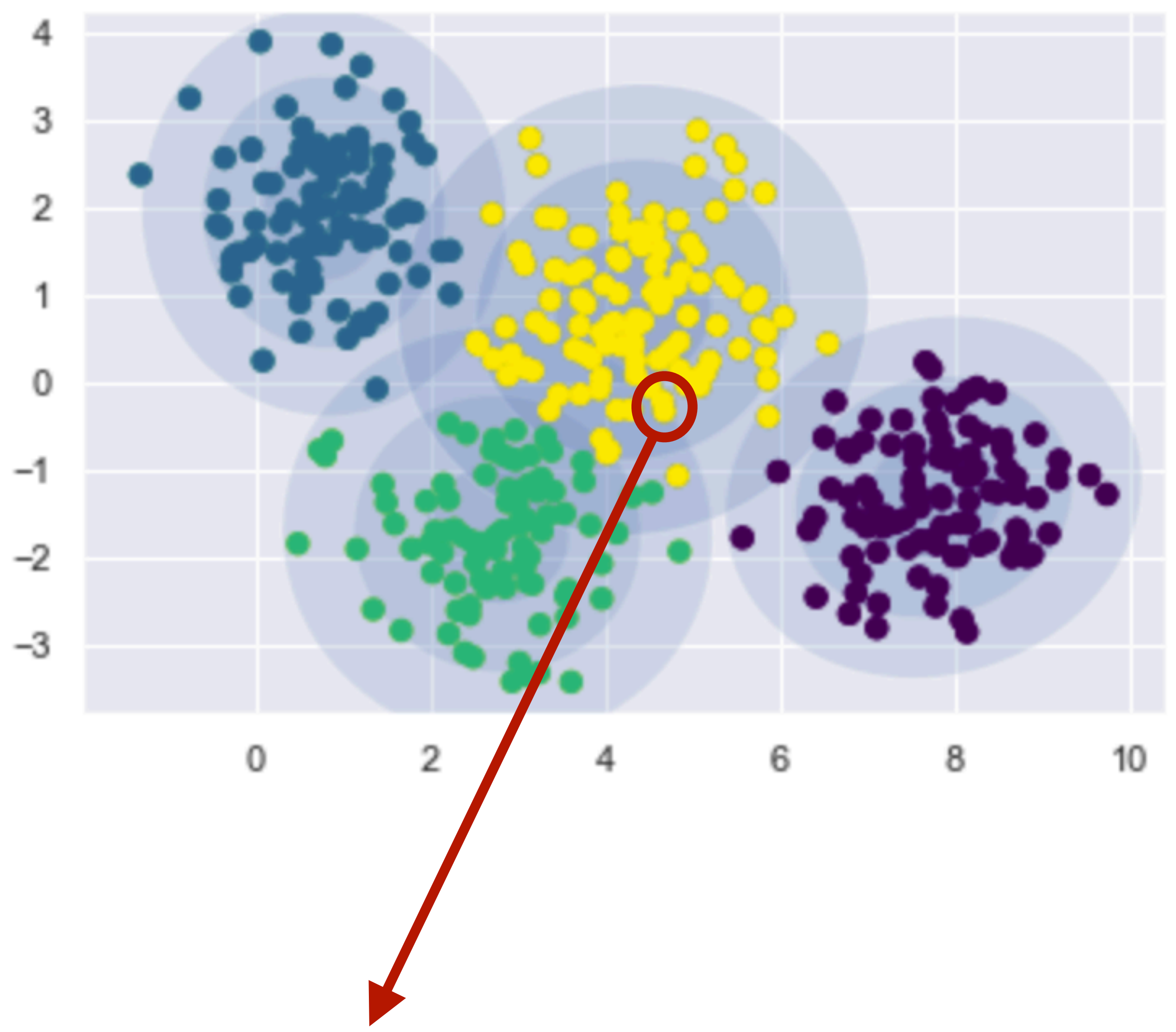
$$p(t=k|\theta)=\pi_k$$

$$p(x|t=k,\theta)=\mathcal{N}(x|\mu_k,\Sigma_k)$$

$$p(x|\theta)=\sum_{k=1}^{4}p(x|t=k,\theta)p(t=k|\theta)$$

2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model



We want to fit a Gaussian Mixture Model !

$$\pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2) + \pi_3 \mathcal{N}(\mu_3, \Sigma_3) + \pi_4 \mathcal{N}(\mu_4, \Sigma_4)$$

parameters : $\theta = \{\pi_1, \mu_1, \Sigma_1, \pi_2, \mu_2, \Sigma_2, \pi_3, \mu_3, \Sigma_3, \pi_4, \mu_4, \Sigma_4\}$

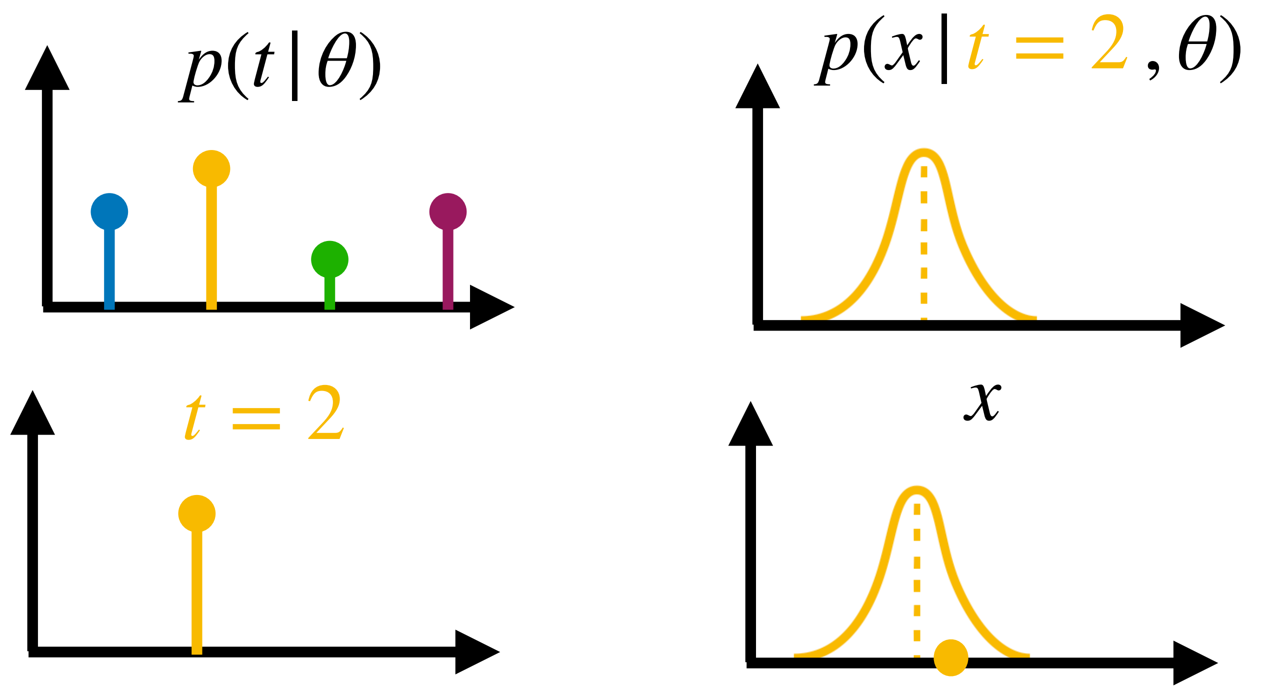
Hard clustering : if we **know the source** of each instances then,

$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{hard}^{MLE}, \Sigma_{hard}^{MLE})$$

$$\mu_{hard}^{MLE} = \frac{\sum_{i \in \text{cluster 2}} x_i}{\text{Number of points in cluster 2}}$$

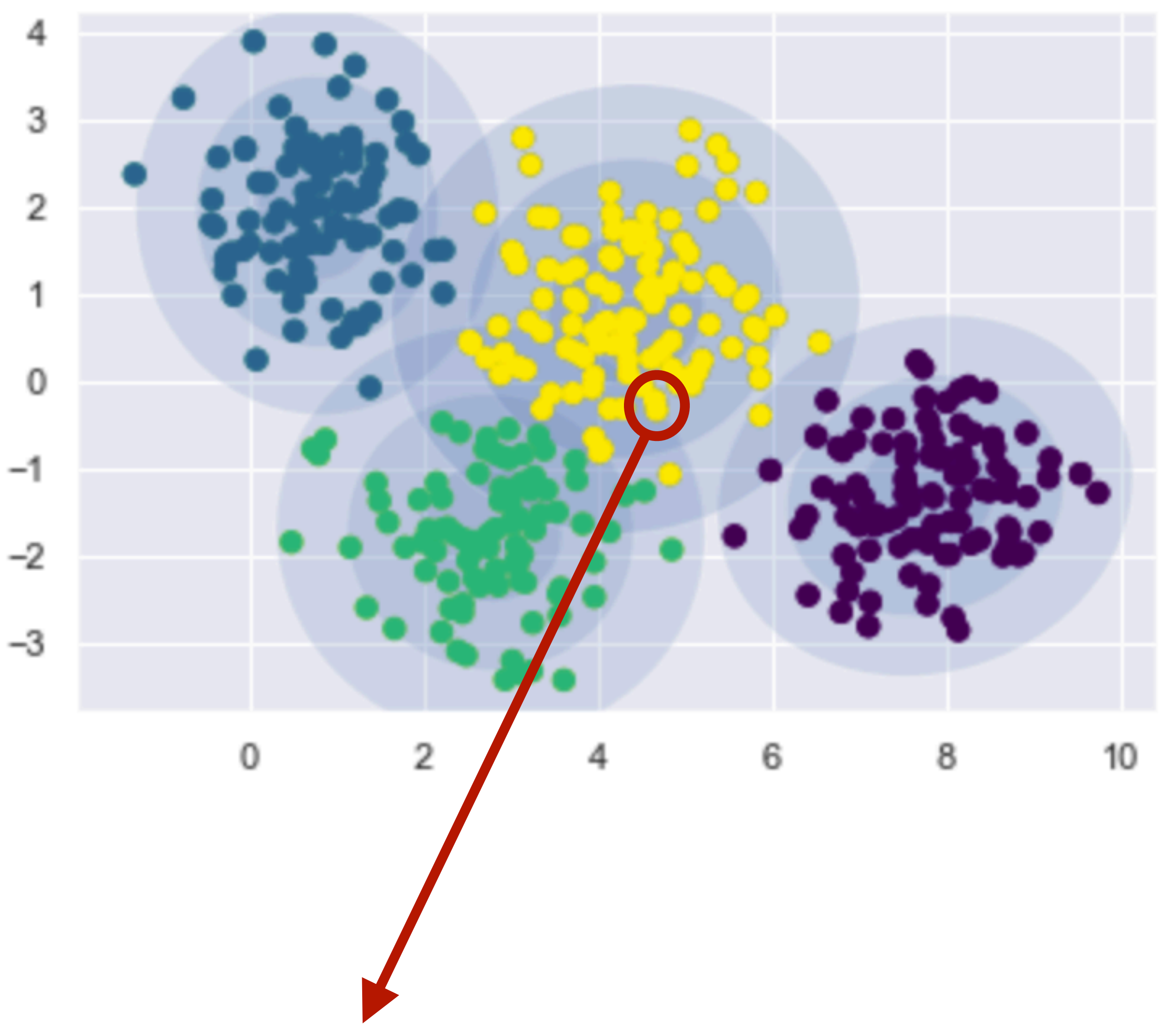
$$\Sigma_{hard}^{MLE} = \frac{\sum_{i \in \text{cluster 2}} (x_i - \mu_{hard}^{MLE}) \times (x_i - \mu_{hard}^{MLE})^T}{\text{Number of points in cluster 2}}$$

We assume that this x is generated as follows :

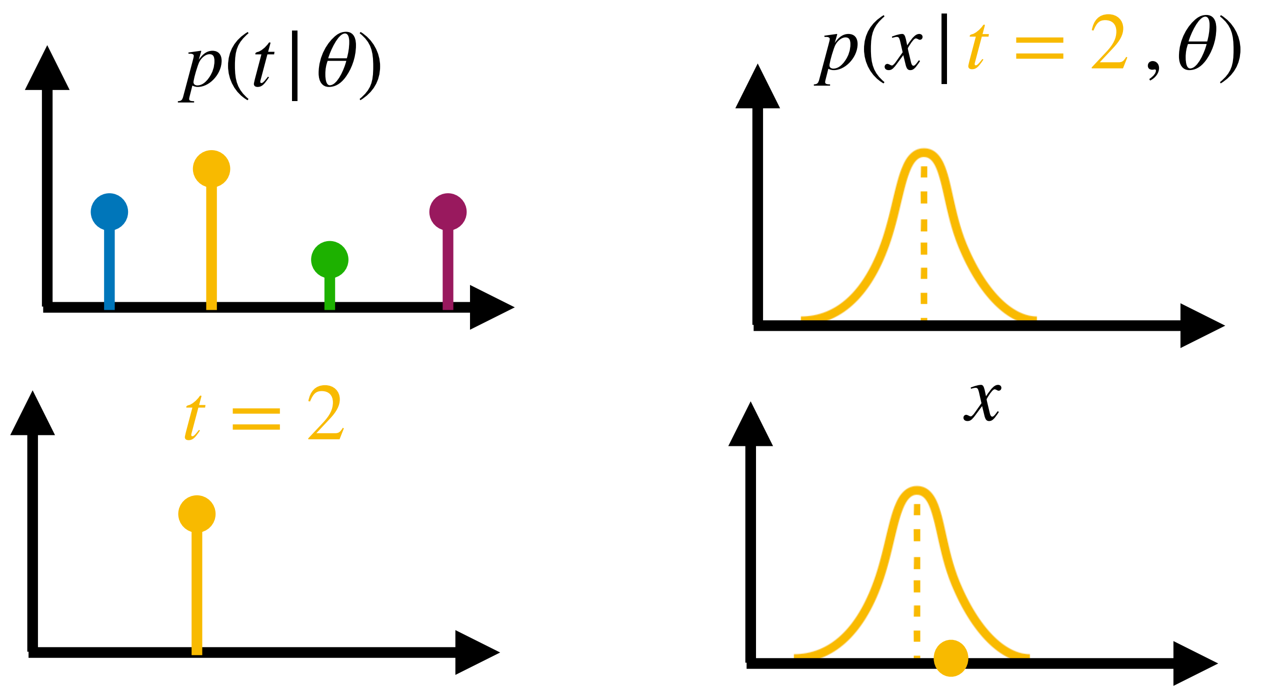


2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model



We assume that this x is generated as follows :



We want to fit a Gaussian Mixture Model !

$$\pi_1\mathcal{N}(\mu_1,\Sigma_1)+\pi_2\mathcal{N}(\mu_2,\Sigma_2)+\pi_3\mathcal{N}(\mu_3,\Sigma_3)+\pi_4\mathcal{N}(\mu_4,\Sigma_4)$$

parameters : $\theta=\{\pi_1,\mu_1,\Sigma_1,\pi_2,\mu_2,\Sigma_2,\pi_3,\mu_3,\Sigma_3,\pi_4,\mu_4,\Sigma_4\}$

Hard clustering : if we **know the source** of each instances then,

$$p(x\,|\,t=2\,,\theta)=\mathcal{N}(x\,|\,\mu_{hard}^{MLE},\Sigma_{hard}^{MLE})$$

$$\mu_{hard}^{MLE}=\frac{\sum_{i\in\text{cluster 2}}x_i}{\text{Number of points in cluster 2}}$$

$$\Sigma_{hard}^{MLE}=\frac{\sum_{i\in\text{cluster 2}}(x_i-\mu_{hard}^{MLE})\times(x_i-\mu_{hard}^{MLE})^T}{\text{Number of points in cluster 2}}$$

Soft / probabilistic clustering : if we **know the source** of each instances then,

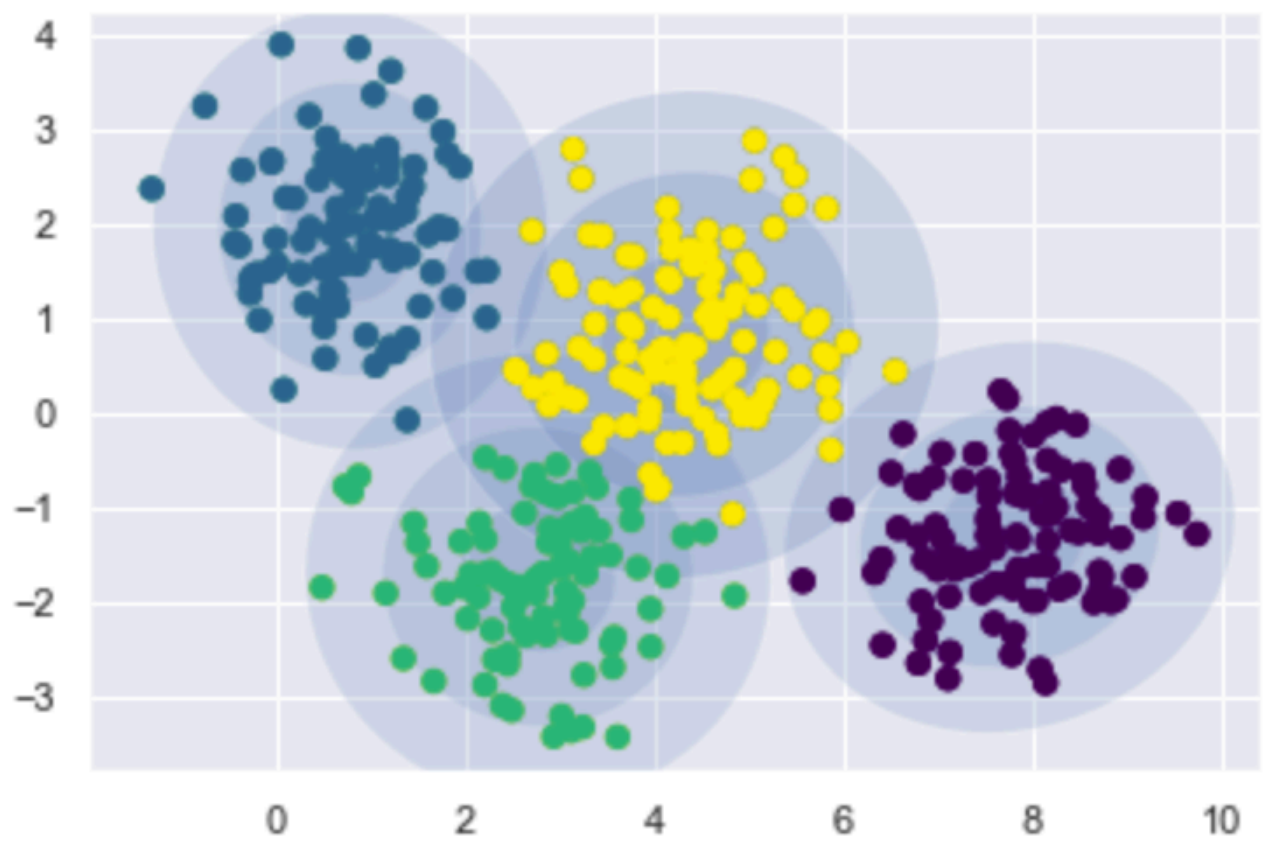
$$p(x\,|\,t=2\,,\theta)=\mathcal{N}(x\,|\,\mu_{soft}^{MLE},\Sigma_{soft}^{MLE})$$

$$\mu_{soft}^{MLE}=\frac{\sum_i\ p(t=2\,|\,x,\theta)\ x_i}{\sum_i p(\ t=2\,|\,x,\theta)}$$

$$\Sigma_{soft}^{MLE}=\frac{\sum_i\ p(t=2\,|\,x_i,\theta)\,(x_i-\mu_{soft}^{MLE})\times(x_i-\mu_{soft}^{MLE})^T}{\sum_i p(\ t=2\,|\,x_i,\theta)}$$

2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [0/6]



Soft / probabilistic clustering : if we **know the source** of each instances then,

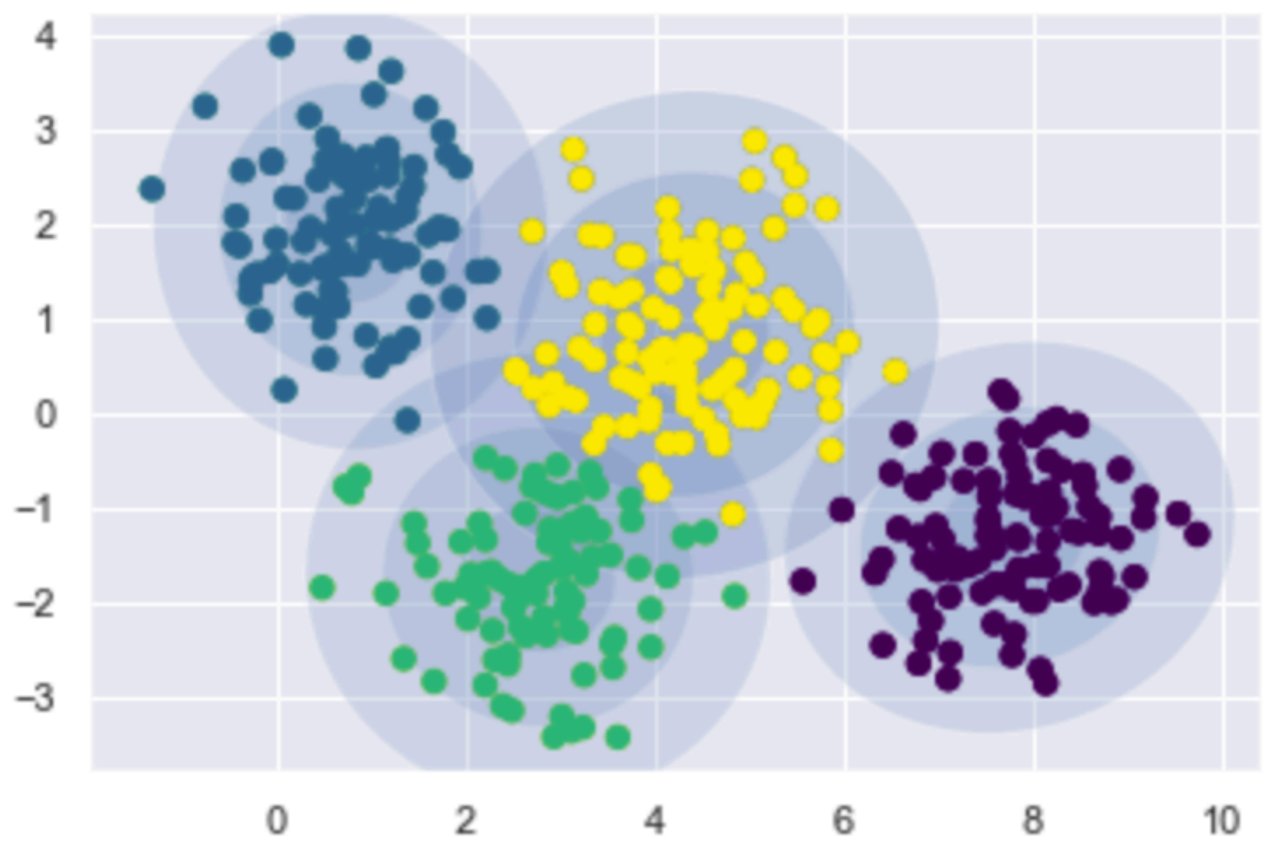
$$p(x \mid t=2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

$$\mu_{soft}^{MLE} = \frac{\sum_i p(t=2 \mid x, \theta)\, x_i}{\sum_i p(t=2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t=2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t=2 \mid x_i, \theta)}$$

2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [0/6]



Soft / probabilistic clustering : if we **know the source** of each instances then,

$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

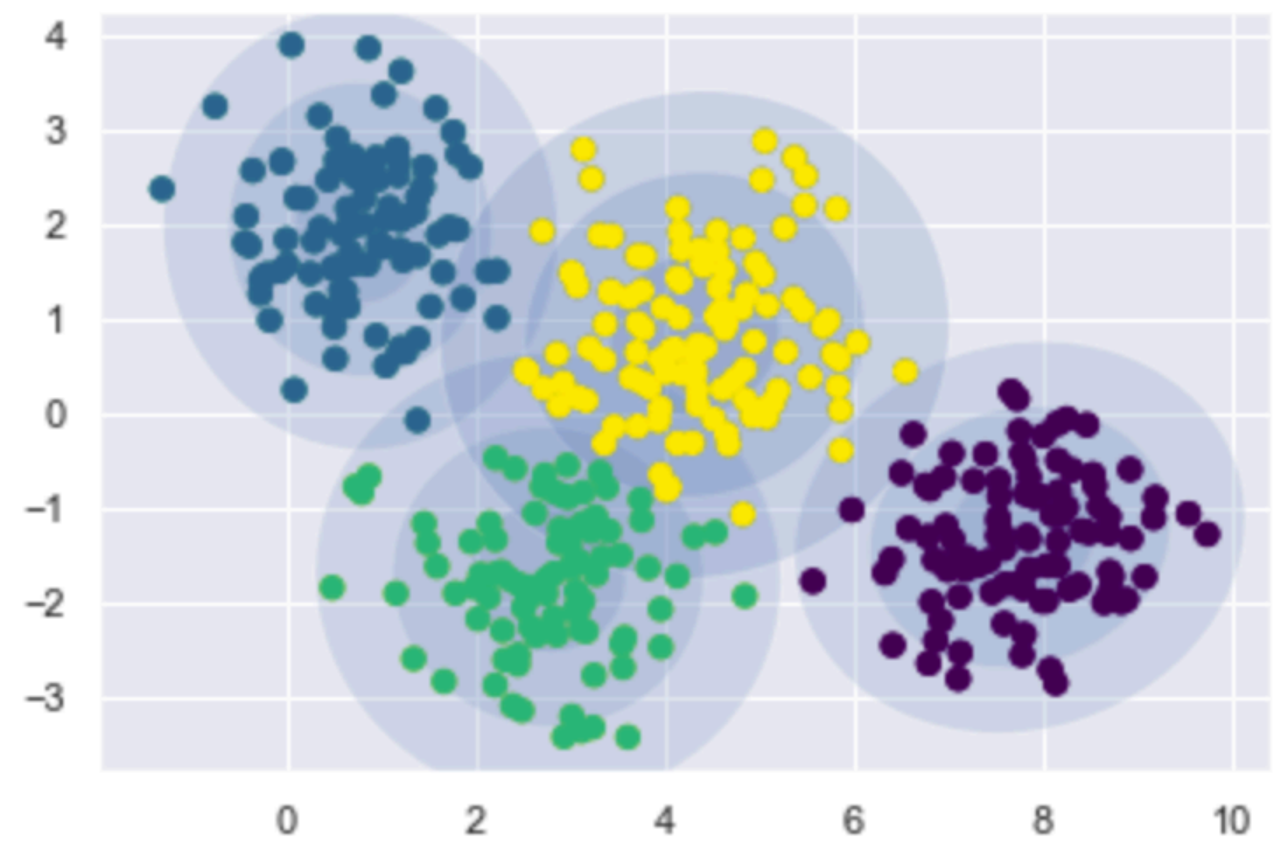
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [0/6]



Soft / probabilistic clustering : if we know the source of each instances then,

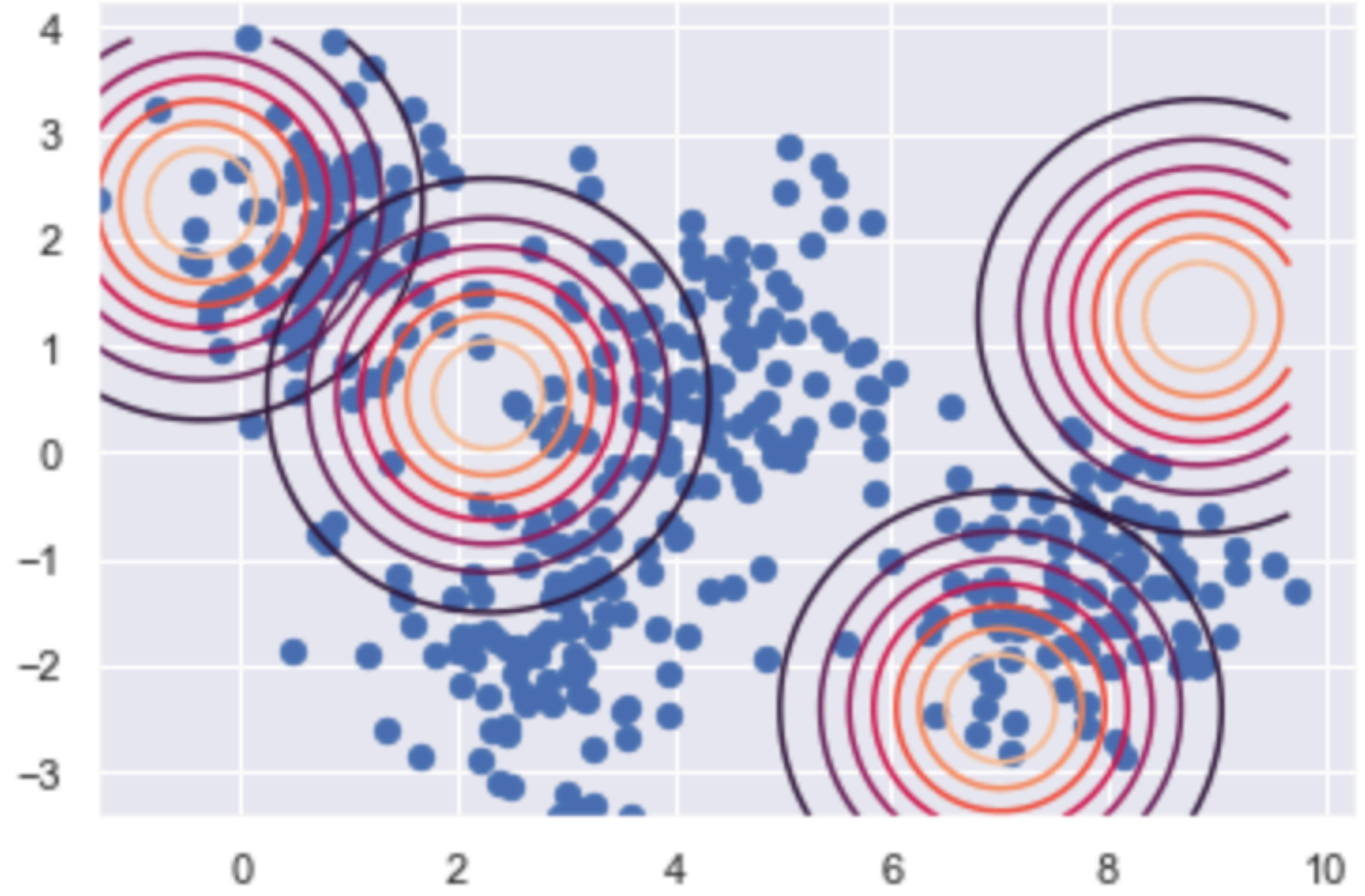
$$p(x \,|\, t=2, \theta) = \mathcal{N}(x \,|\, \mu^{MLE}_{soft}, \Sigma^{MLE}_{soft})$$

$$\mu^{MLE}_{soft} = \frac{\sum_i p(t=2 \,|\, x, \theta)\, x_i}{\sum_i p(t=2 \,|\, x, \theta)}$$

$$\Sigma^{MLE}_{soft} = \frac{\sum_i p(t=2 \,|\, x_i, \theta)\,(x_i - \mu^{MLE}_{soft}) \times (x_i - \mu^{MLE}_{soft})^T}{\sum_i p(t=2 \,|\, x_i, \theta)}$$

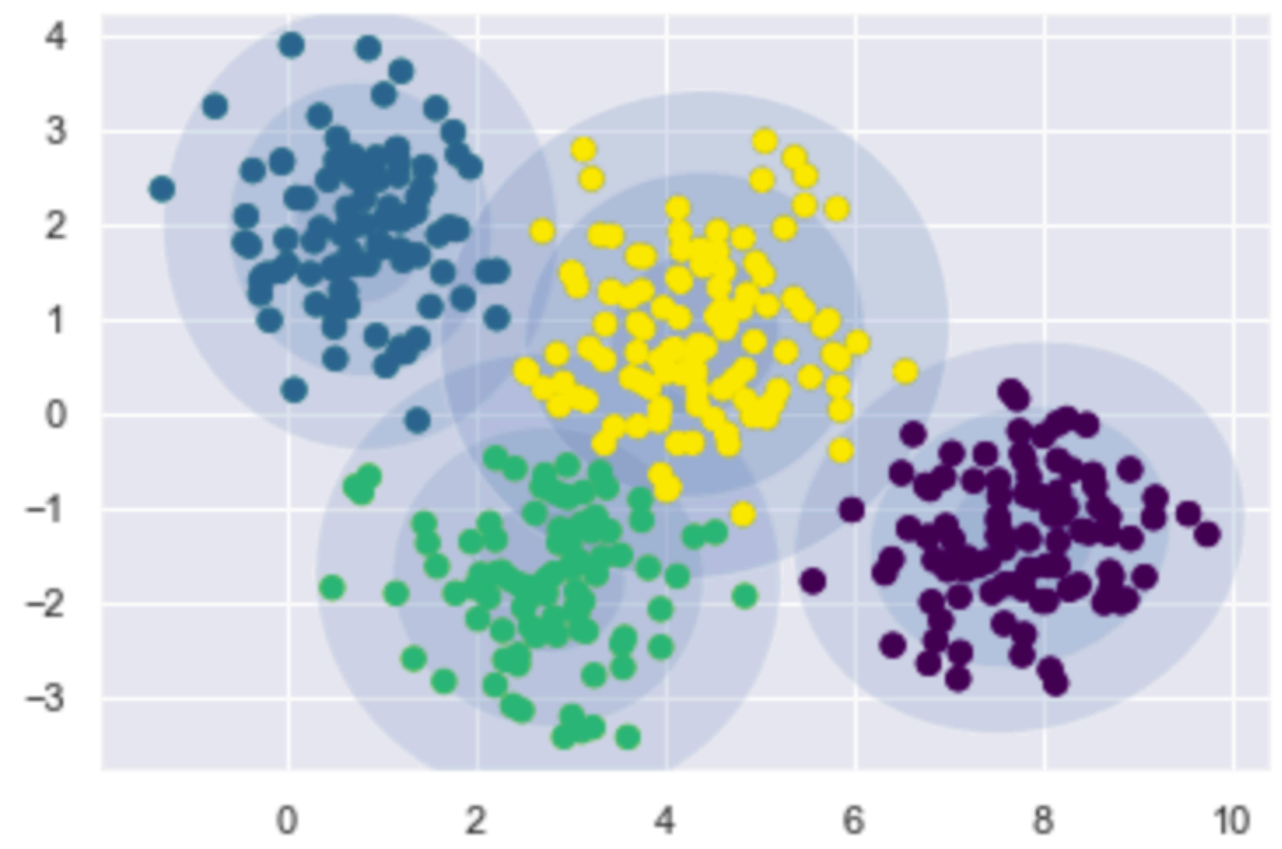
We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

INITIALISATION : first estimation



2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [1/6]



Soft / probabilistic clustering : if we **know the source** of each instances then,

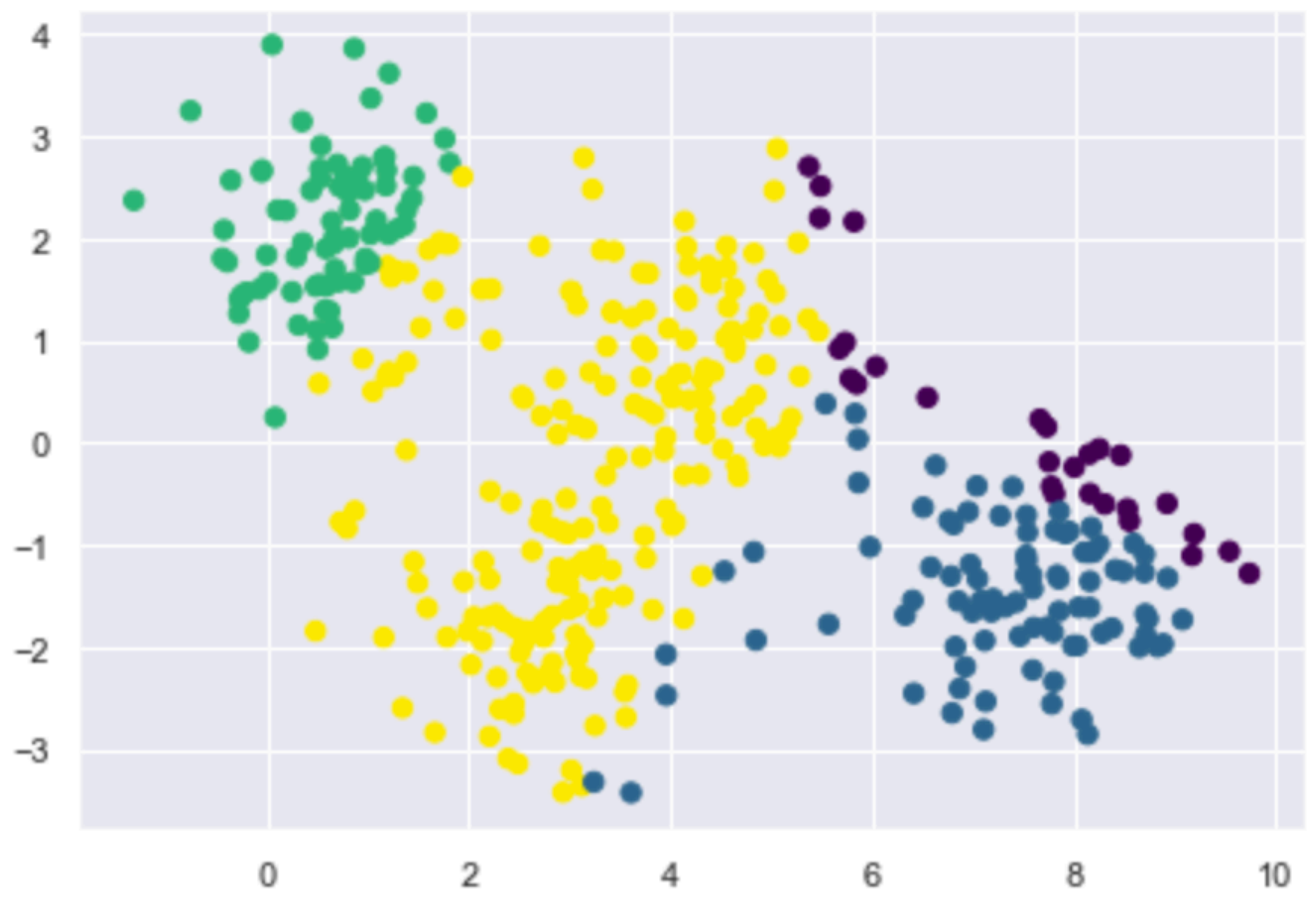
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

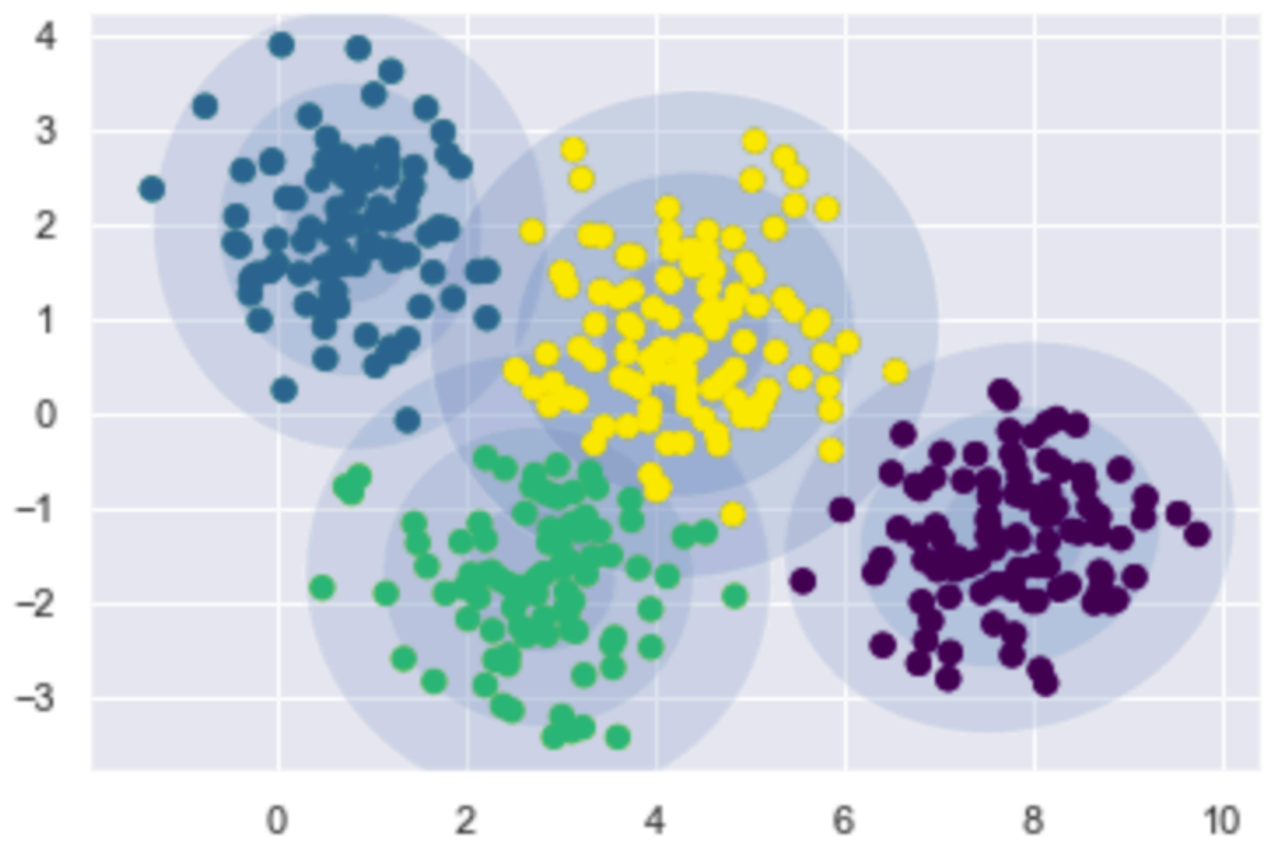
We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

STEP 1



2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [1/6]



Soft / probabilistic clustering : if we **know the source** of each instances then,

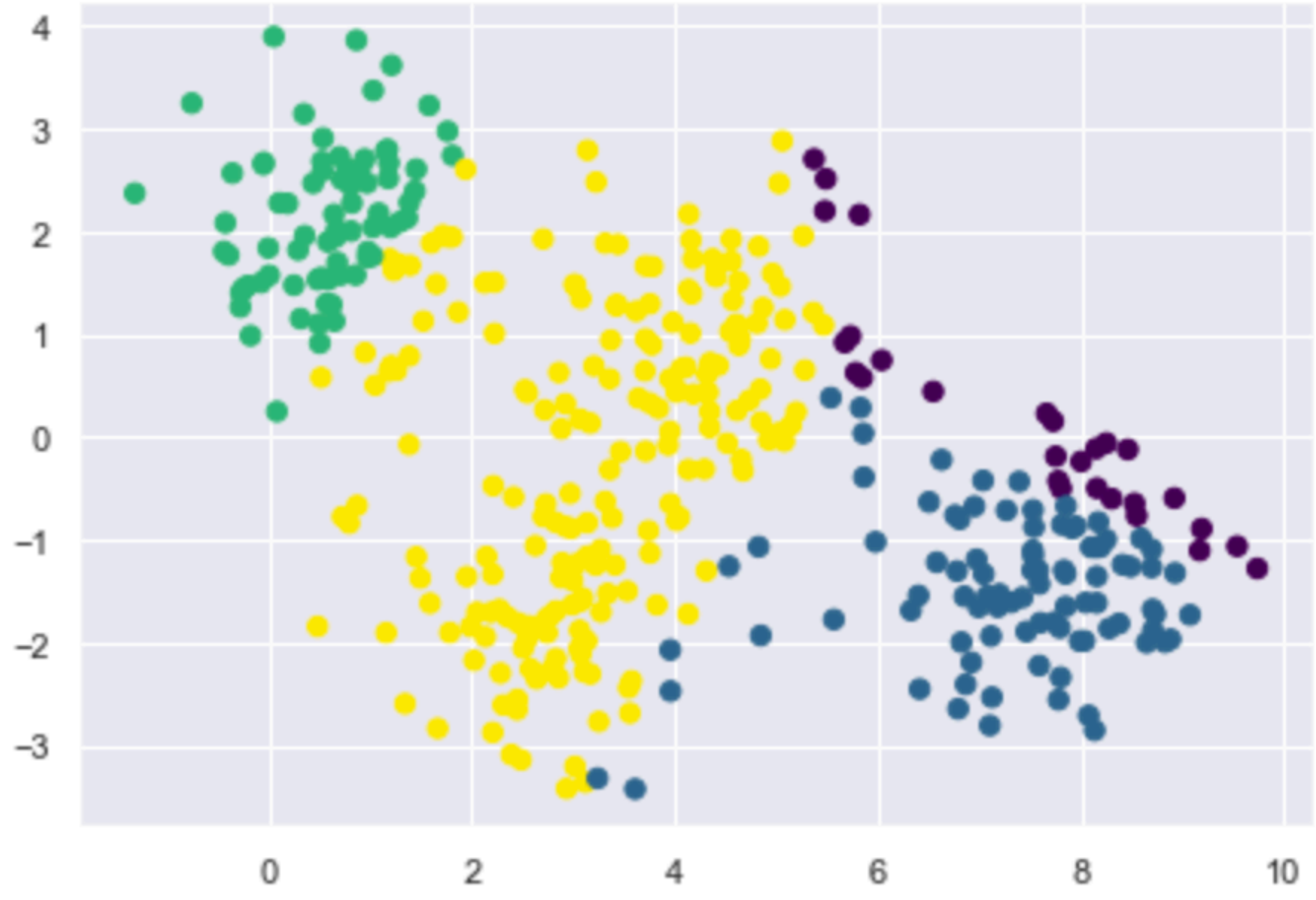
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

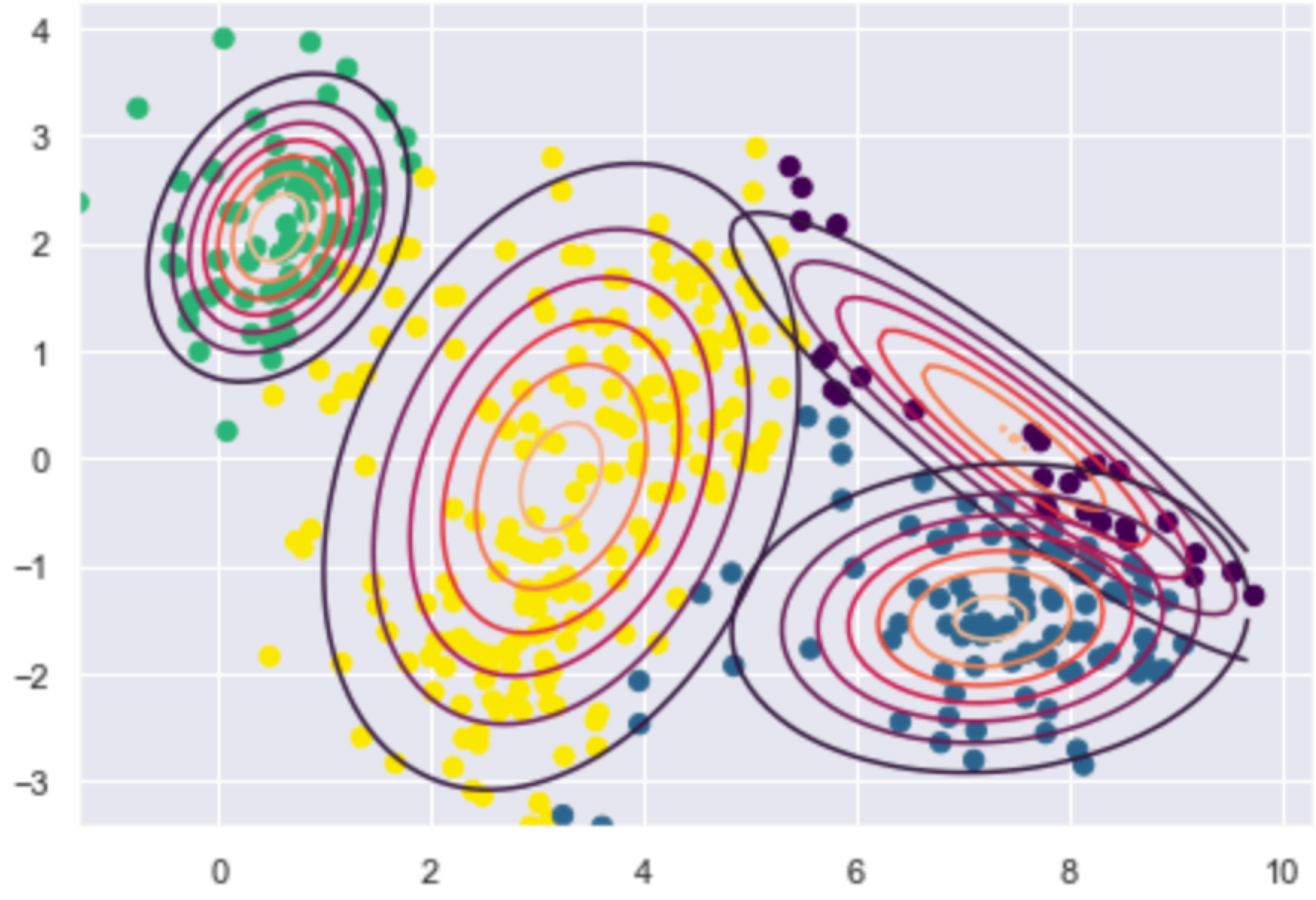
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

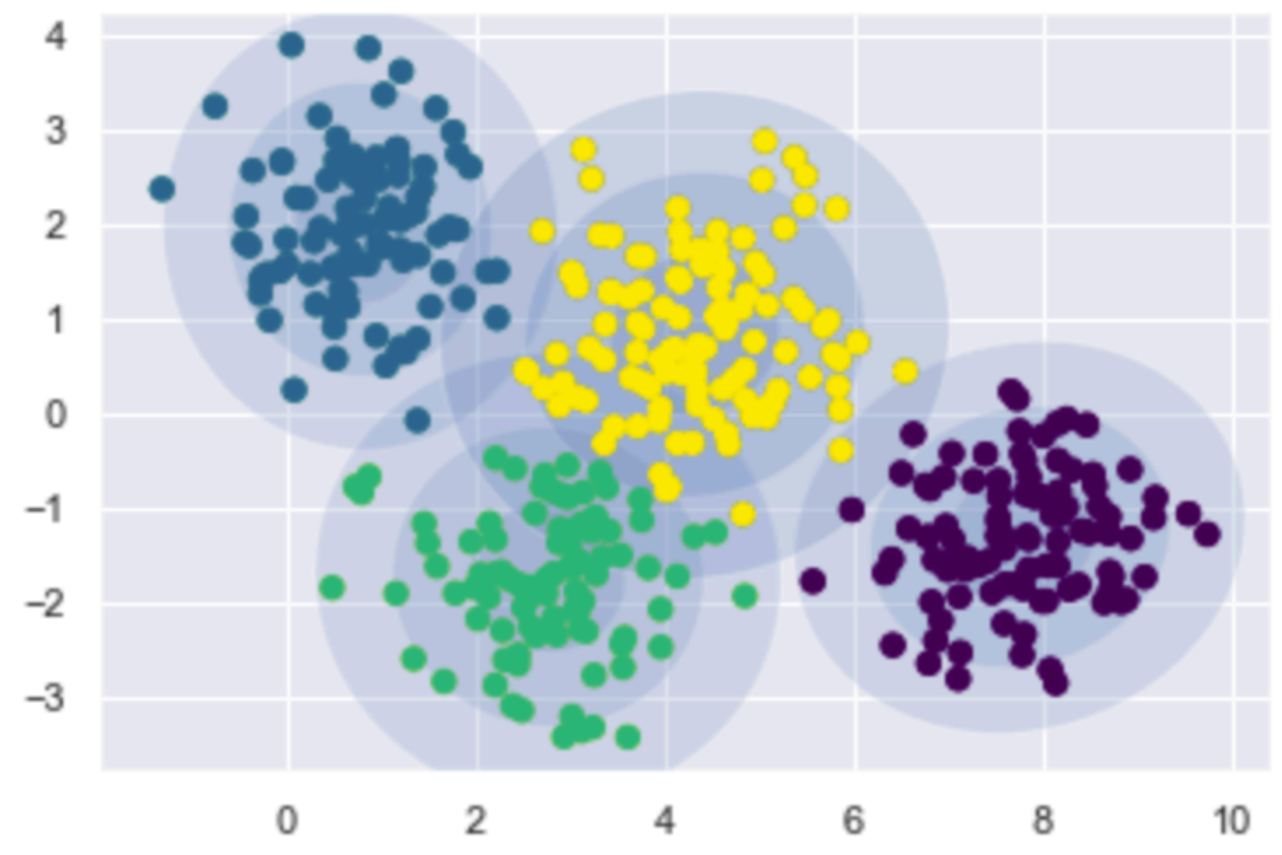
STEP 1





2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [2/6]



Soft / probabilistic clustering : ** if we **know the source of each instances then,

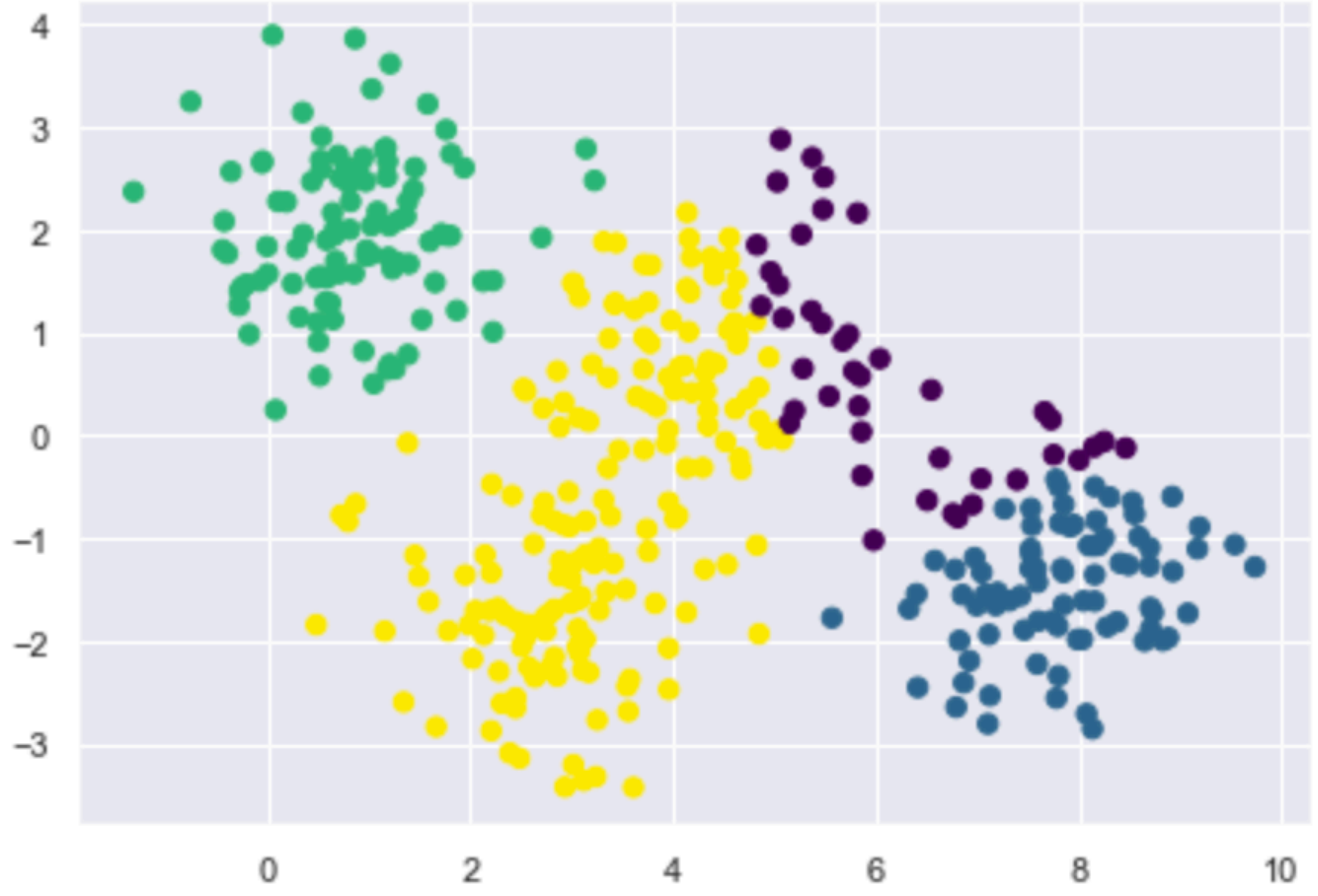
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

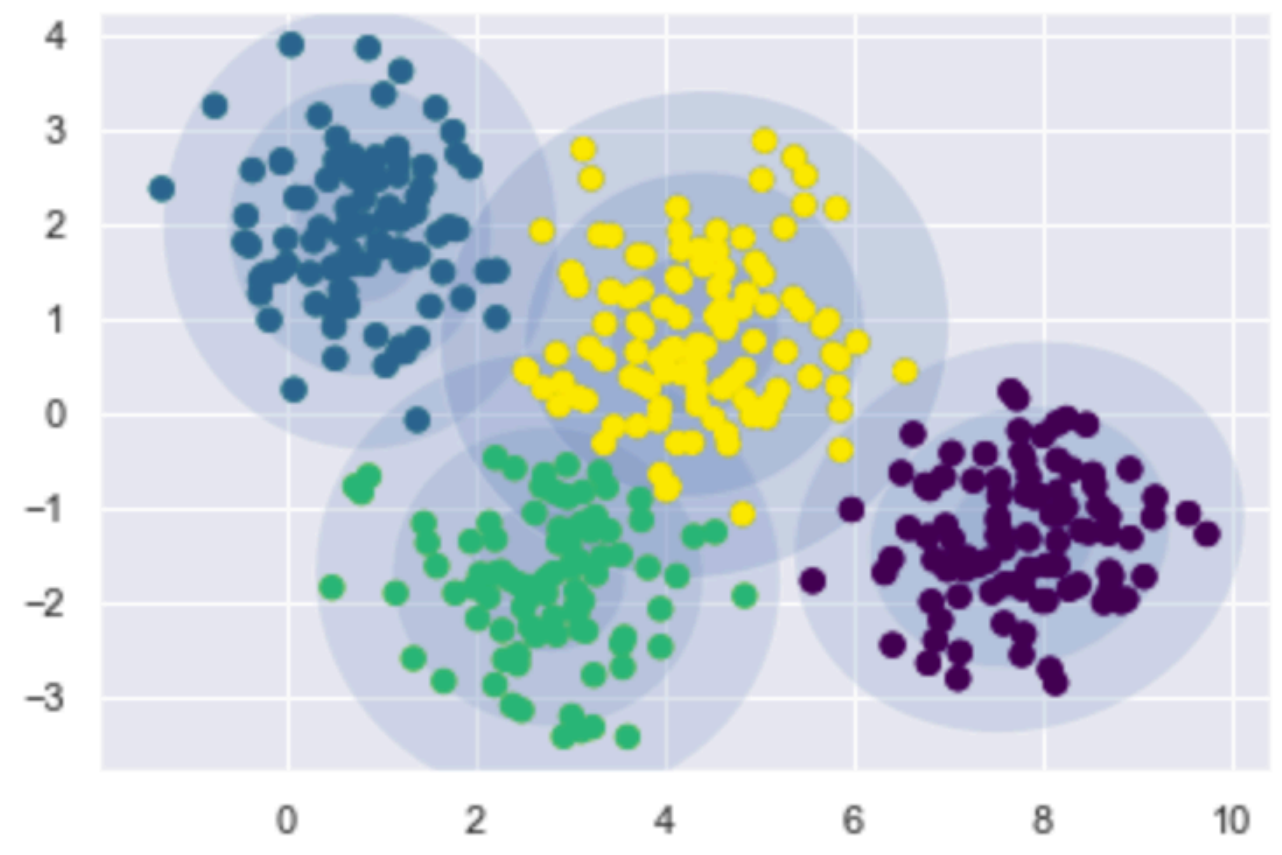
We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

STEP 2



2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [2/6]



Soft / probabilistic clustering : if we know the source of each instances then,

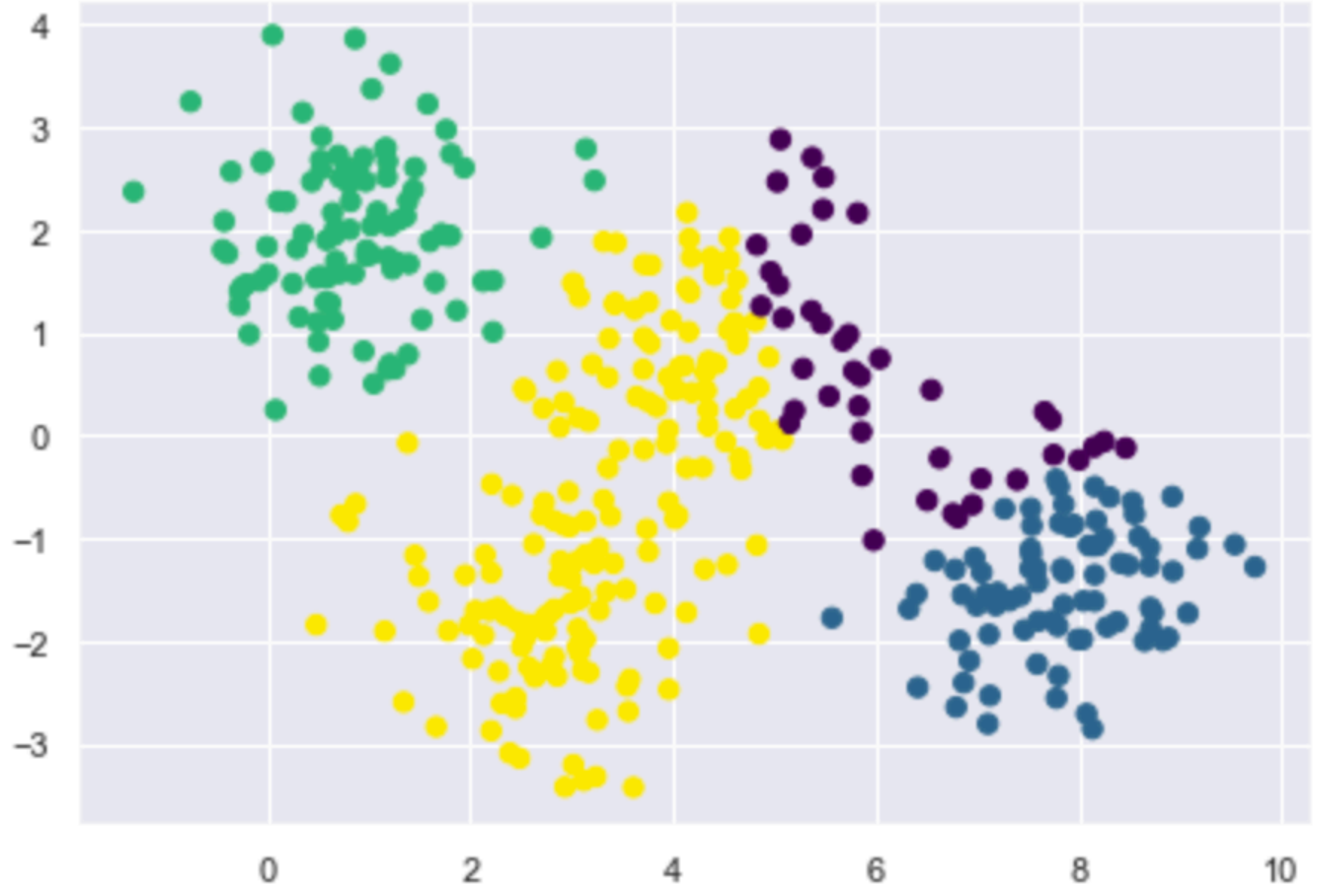
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu^{MLE}_{soft}, \Sigma^{MLE}_{soft})$$

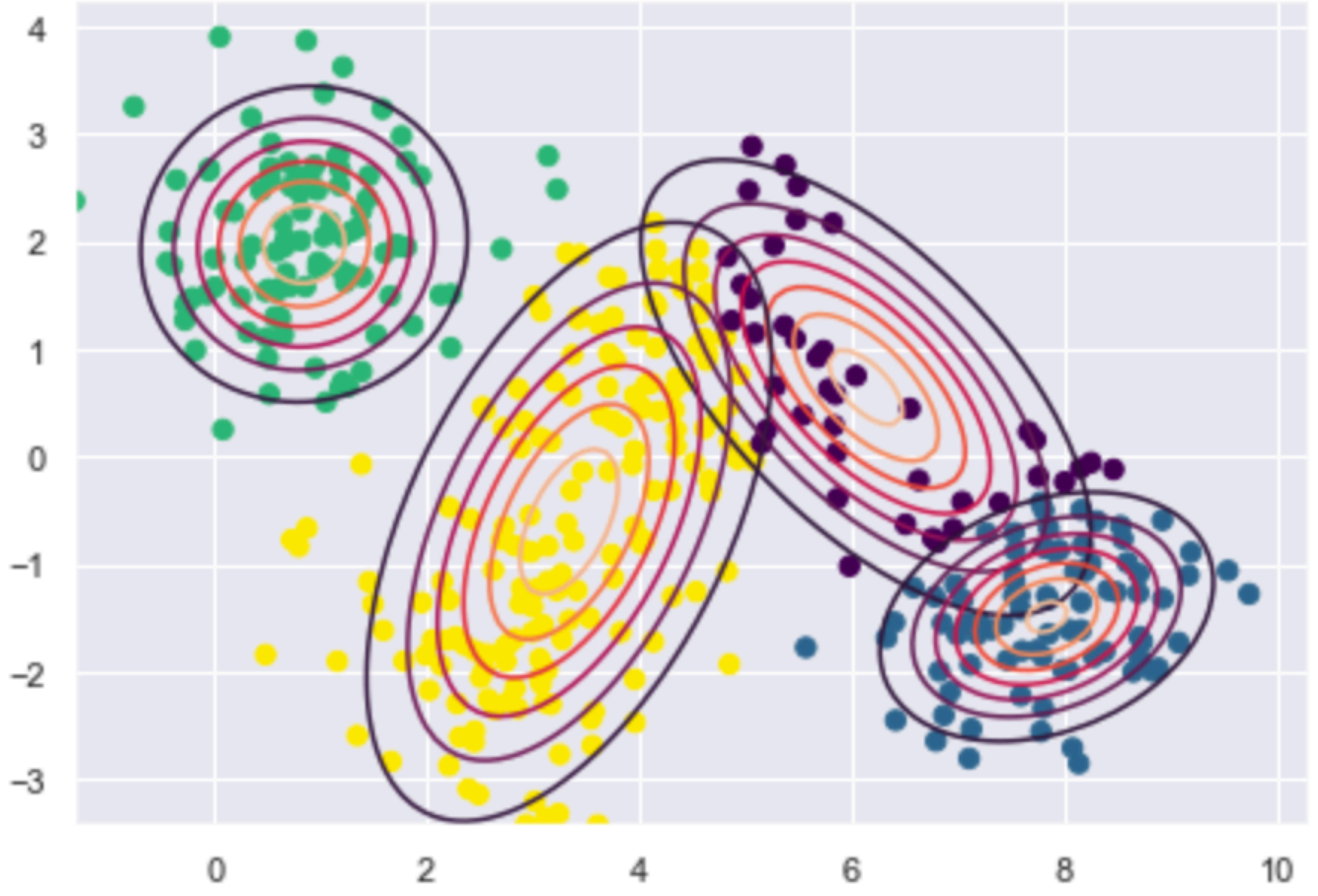
$$\mu^{MLE}_{soft} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma^{MLE}_{soft} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu^{MLE}_{soft}) \times (x_i - \mu^{MLE}_{soft})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

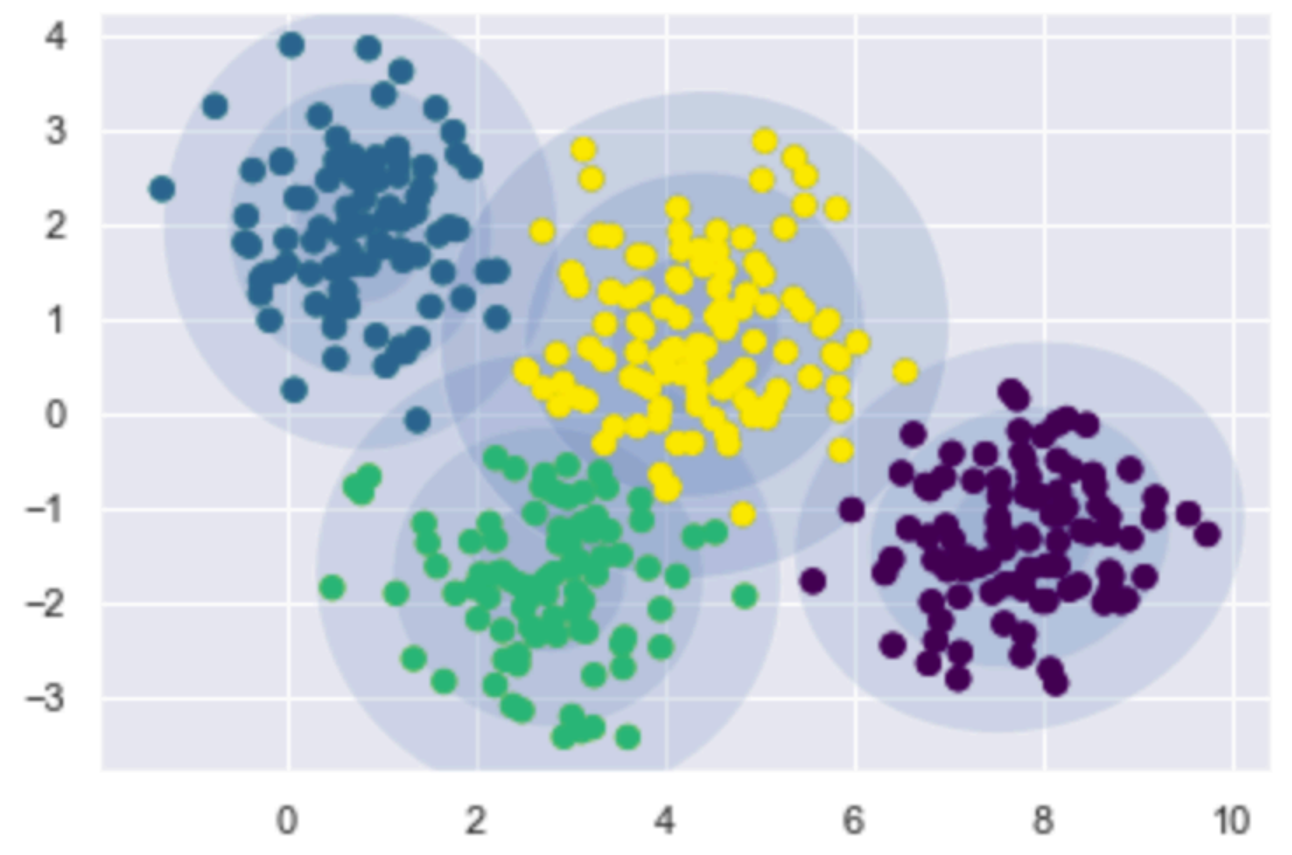
STEP 2





2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [3/6]



Soft / probabilistic clustering : ** if we **know the source of each instances then,

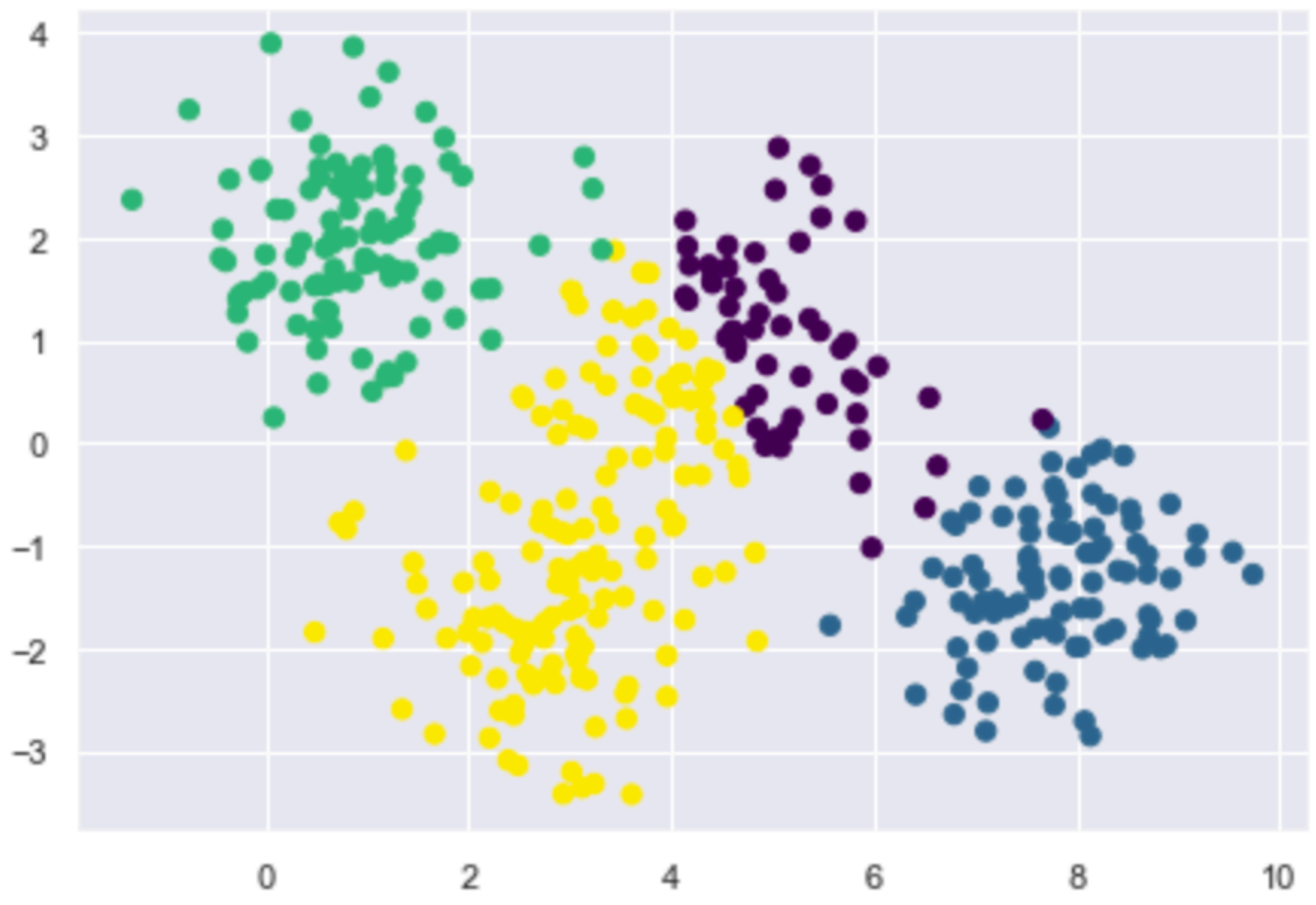
$$p(x \,|\, t = 2, \theta) = \mathcal{N}(x \,|\, \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \,|\, x, \theta)\, x_i}{\sum_i p(t = 2 \,|\, x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \,|\, x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \,|\, x_i, \theta)}$$

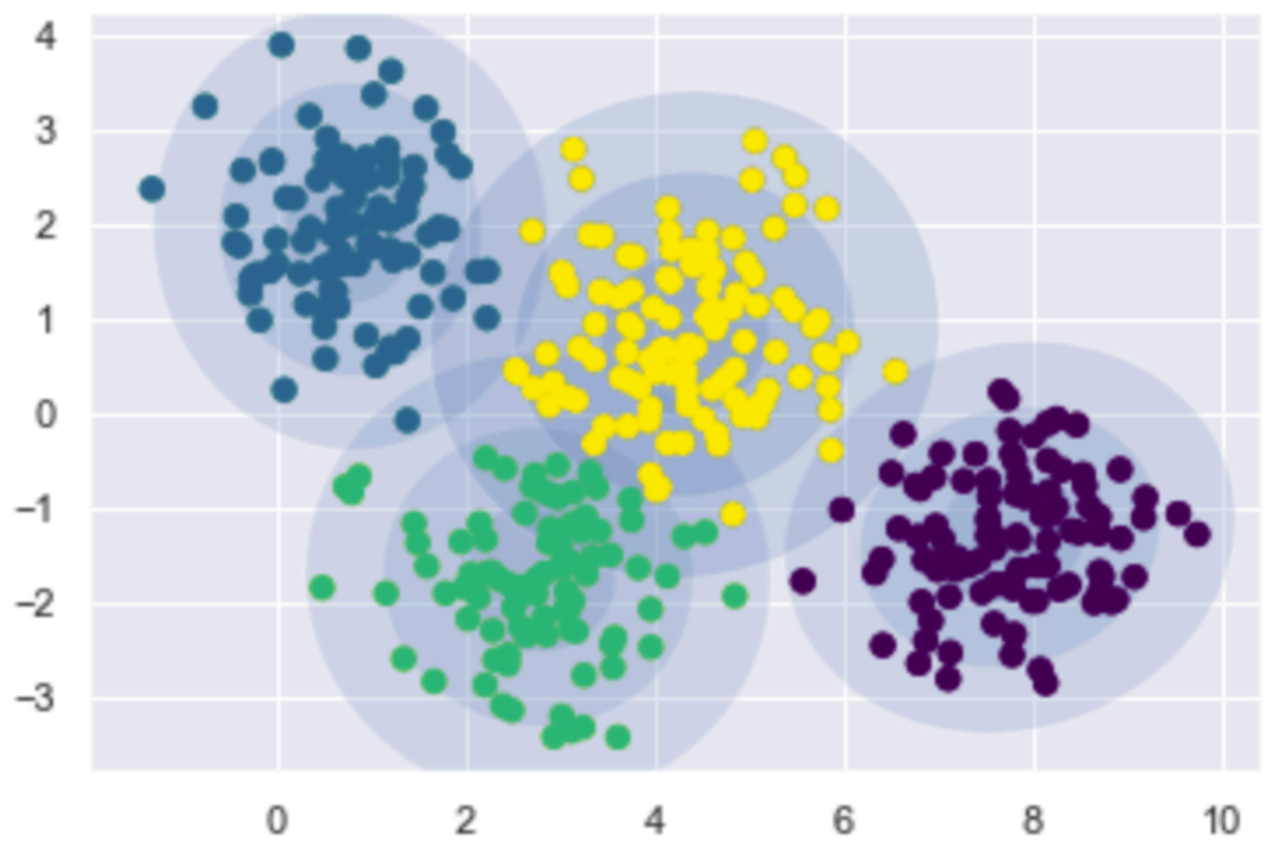
We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

STEP 3



2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [3/6]



Soft / probabilistic clustering : if we **know the source** of each instances then,

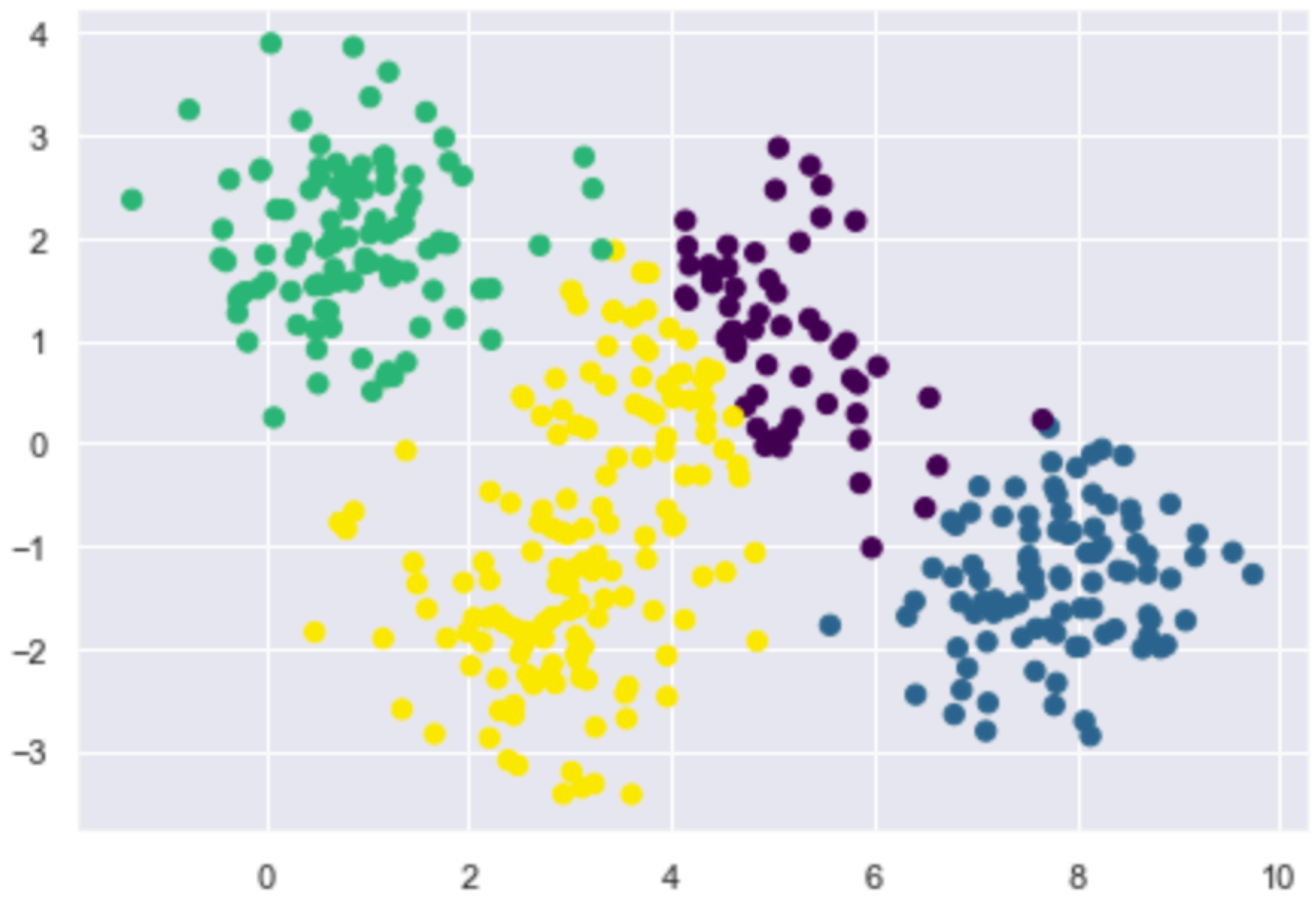
$$p(x \mid t=2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

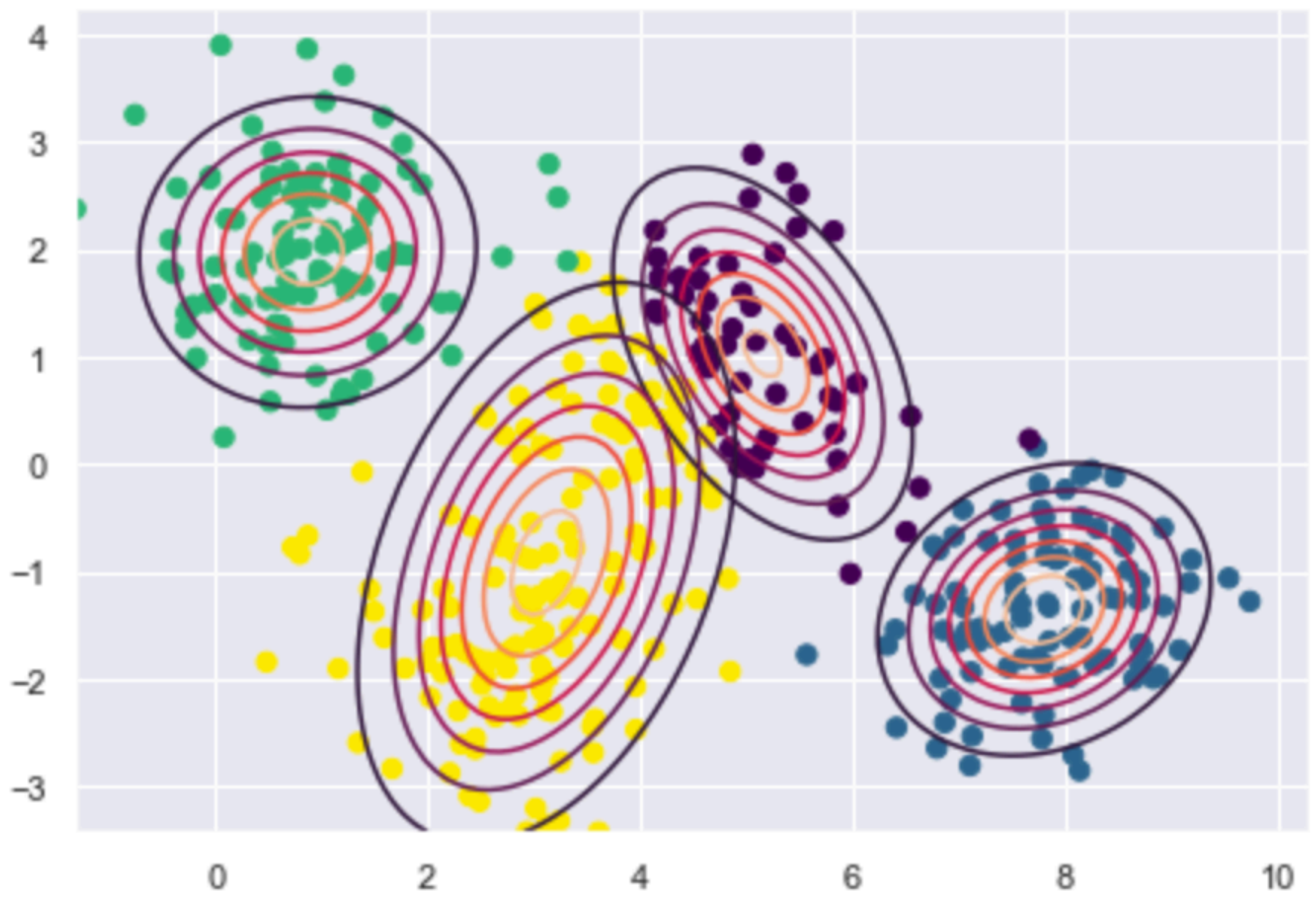
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t=2 \mid x, \theta)\, x_i}{\sum_i p(t=2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t=2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t=2 \mid x_i, \theta)}$$

We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

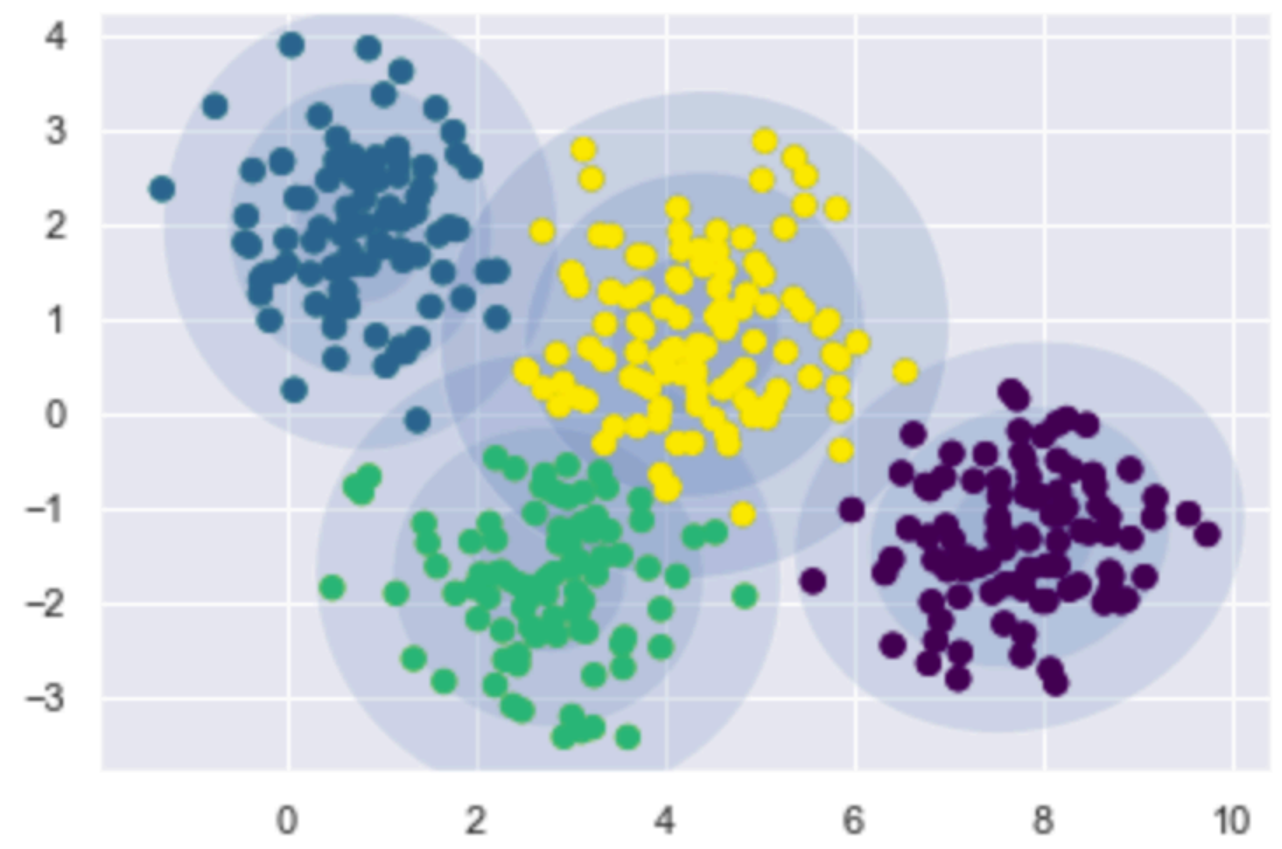
STEP 3





2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [4/6]



Soft / probabilistic clustering : if we **know the source** of each instances then,

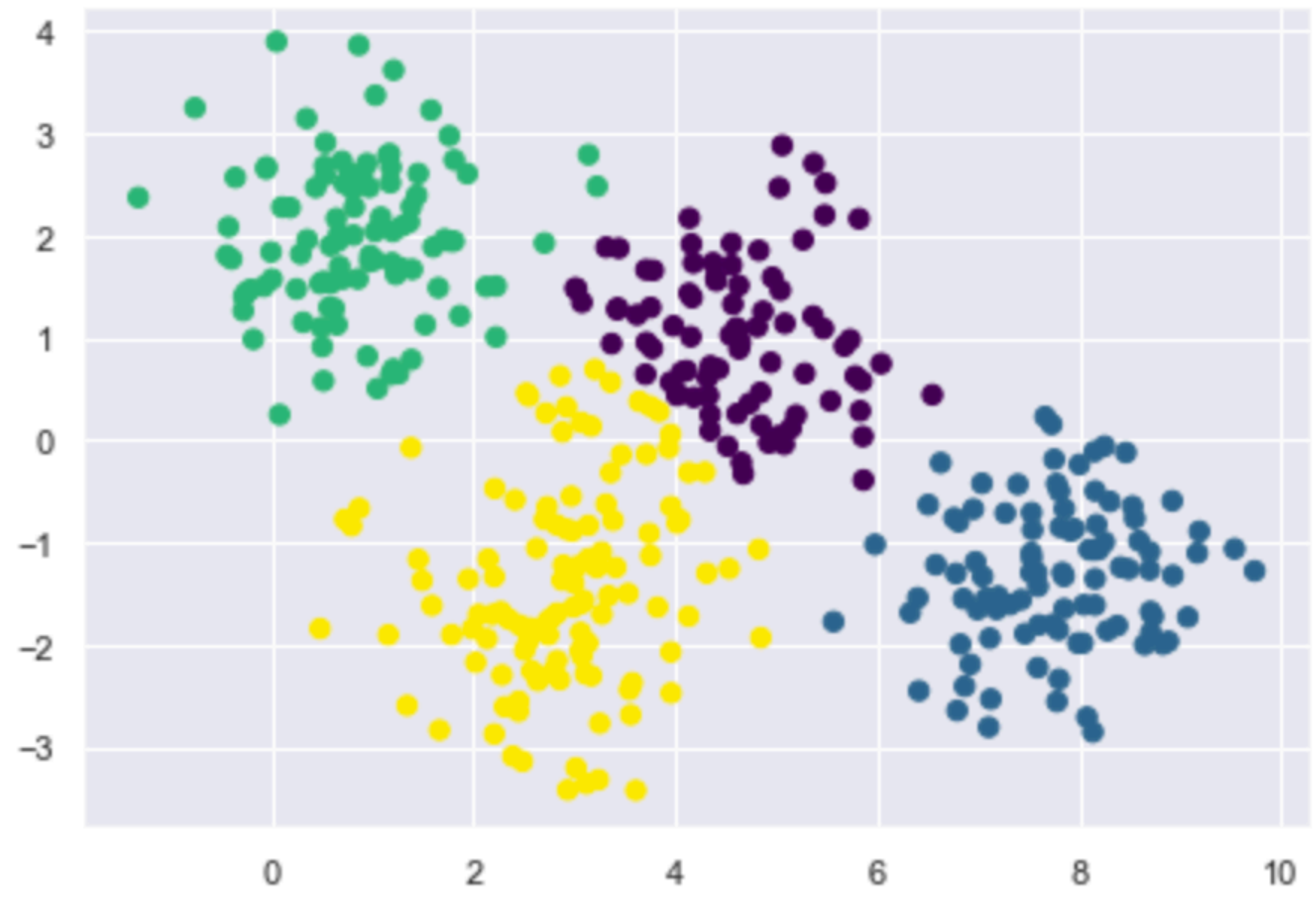
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\, (x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

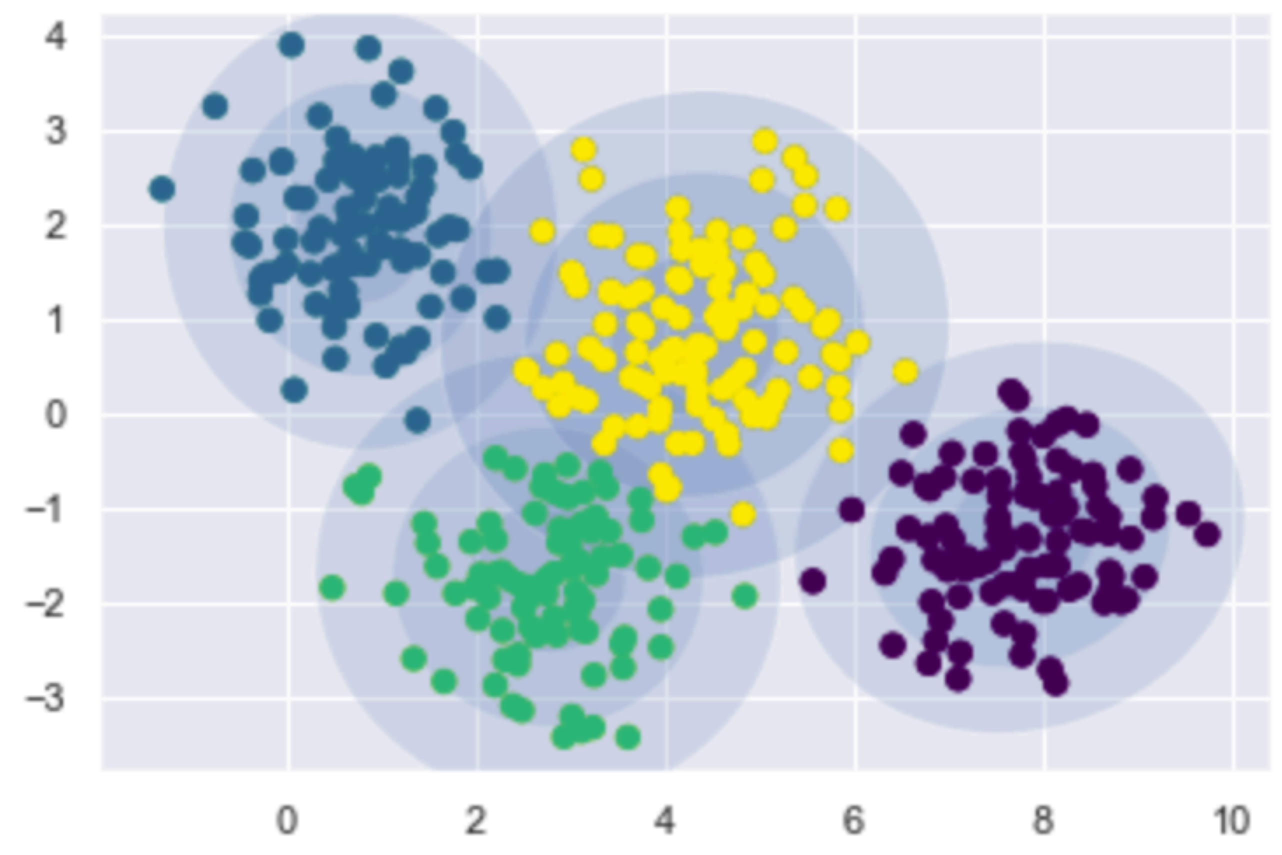
We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

STEP 4



2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [4/6]



Soft / probabilistic clustering : if we **know the source** of each instances then,

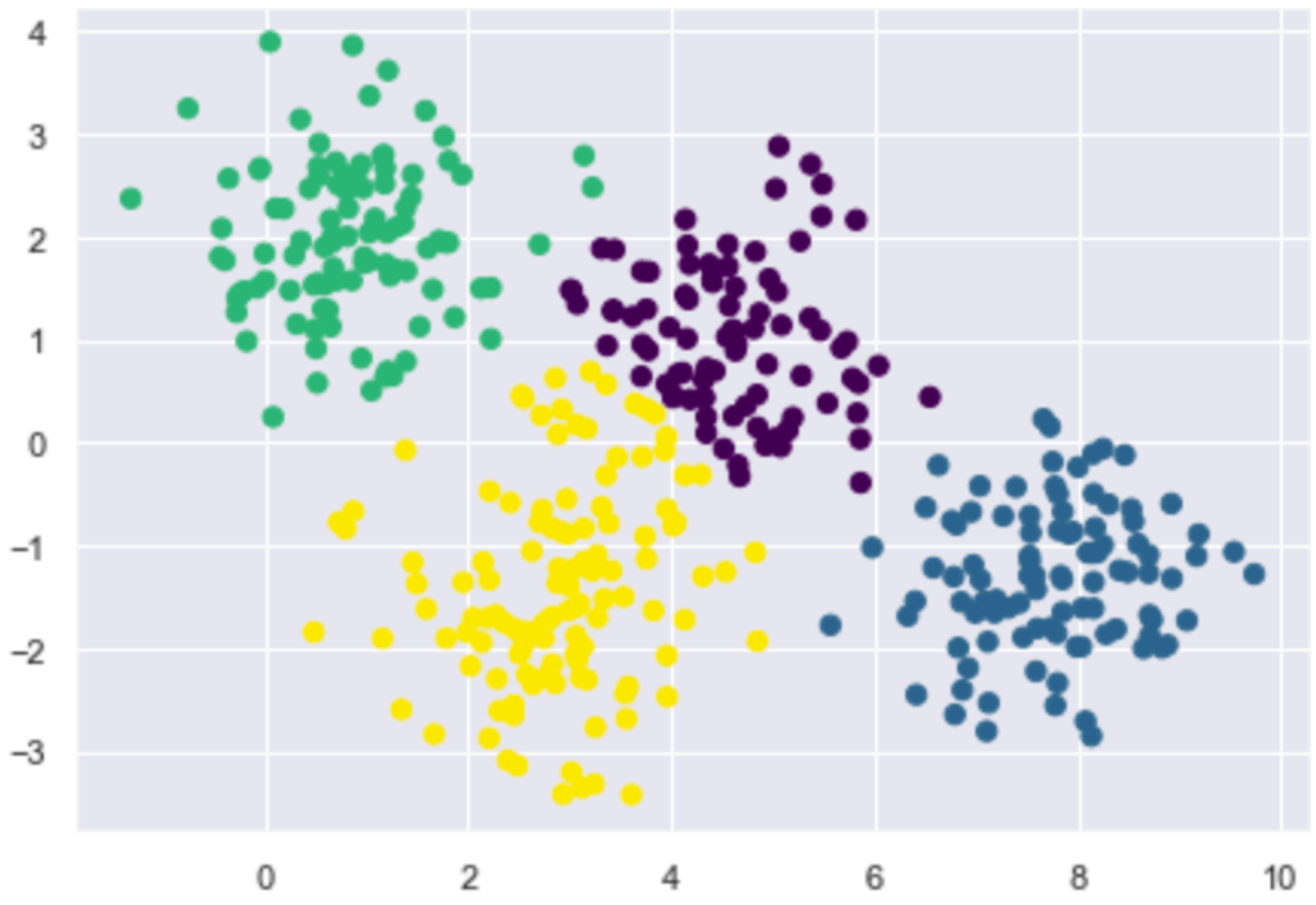
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

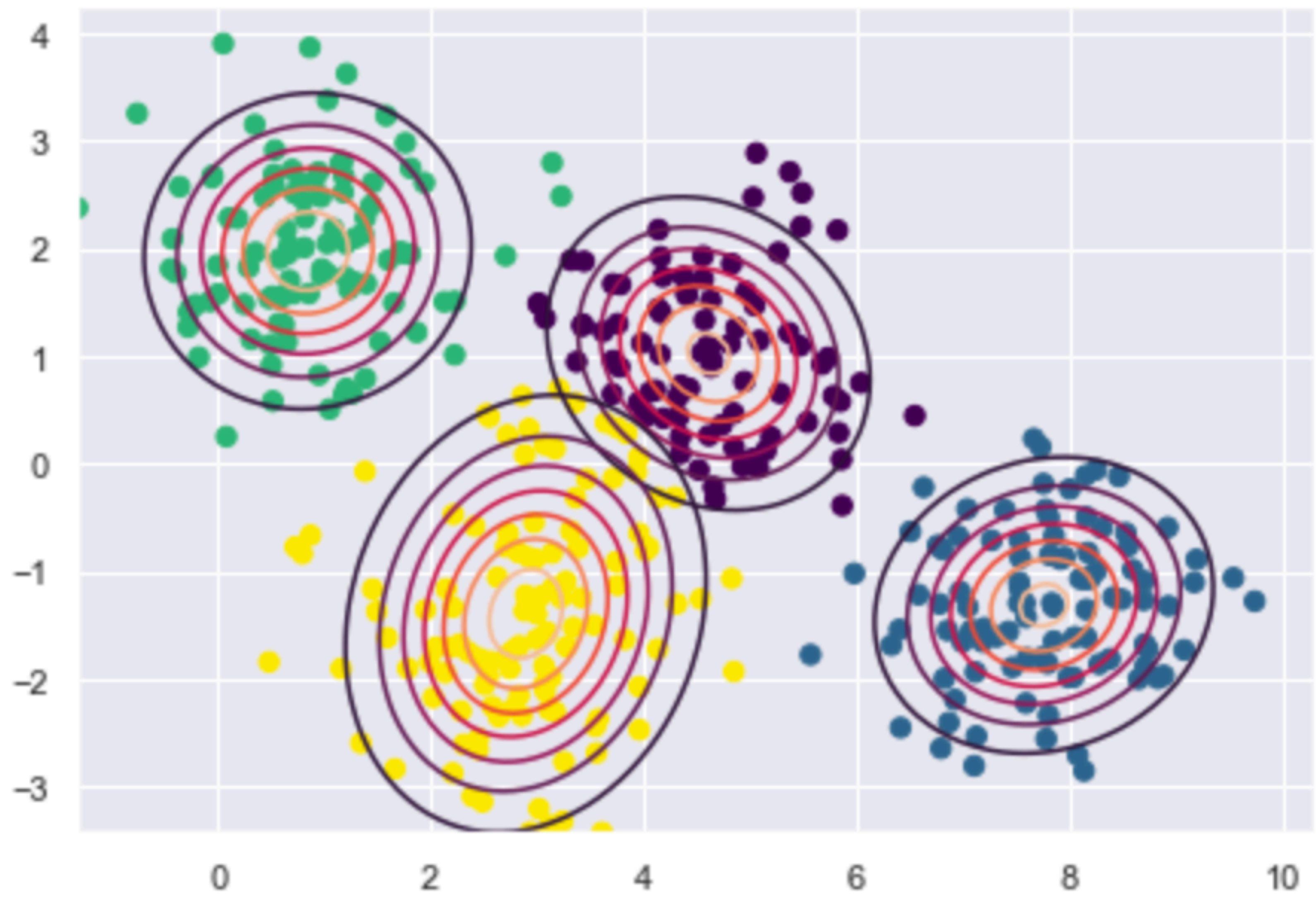
$$\mu_{soft}^{MLE} = \frac{\sum_i \; p(t = 2 \mid x, \theta)\; x_i}{\sum_i p(\; t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i \; p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(\; t = 2 \mid x_i, \theta)}$$

We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

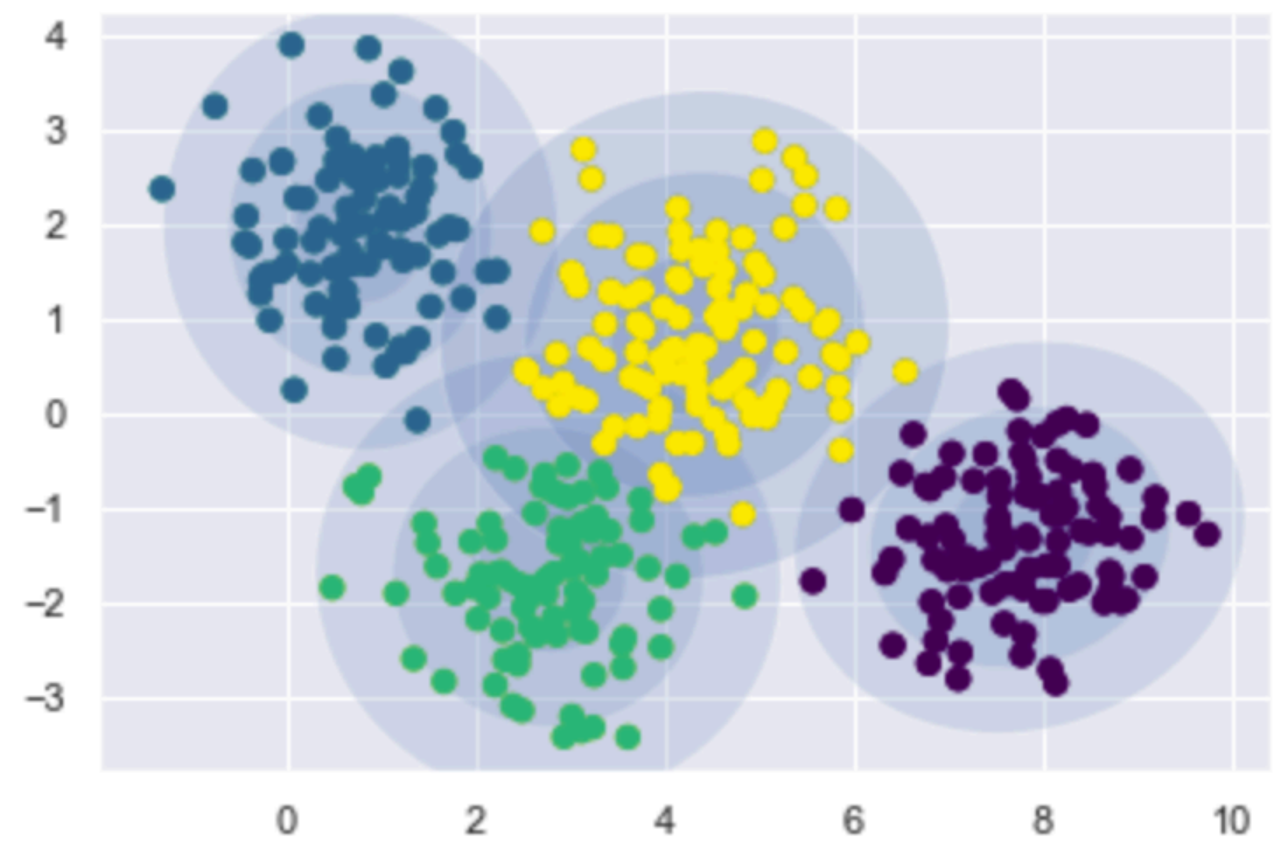
STEP 4





2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [5/6]



Soft / probabilistic clustering : if we know the source of each instances then,

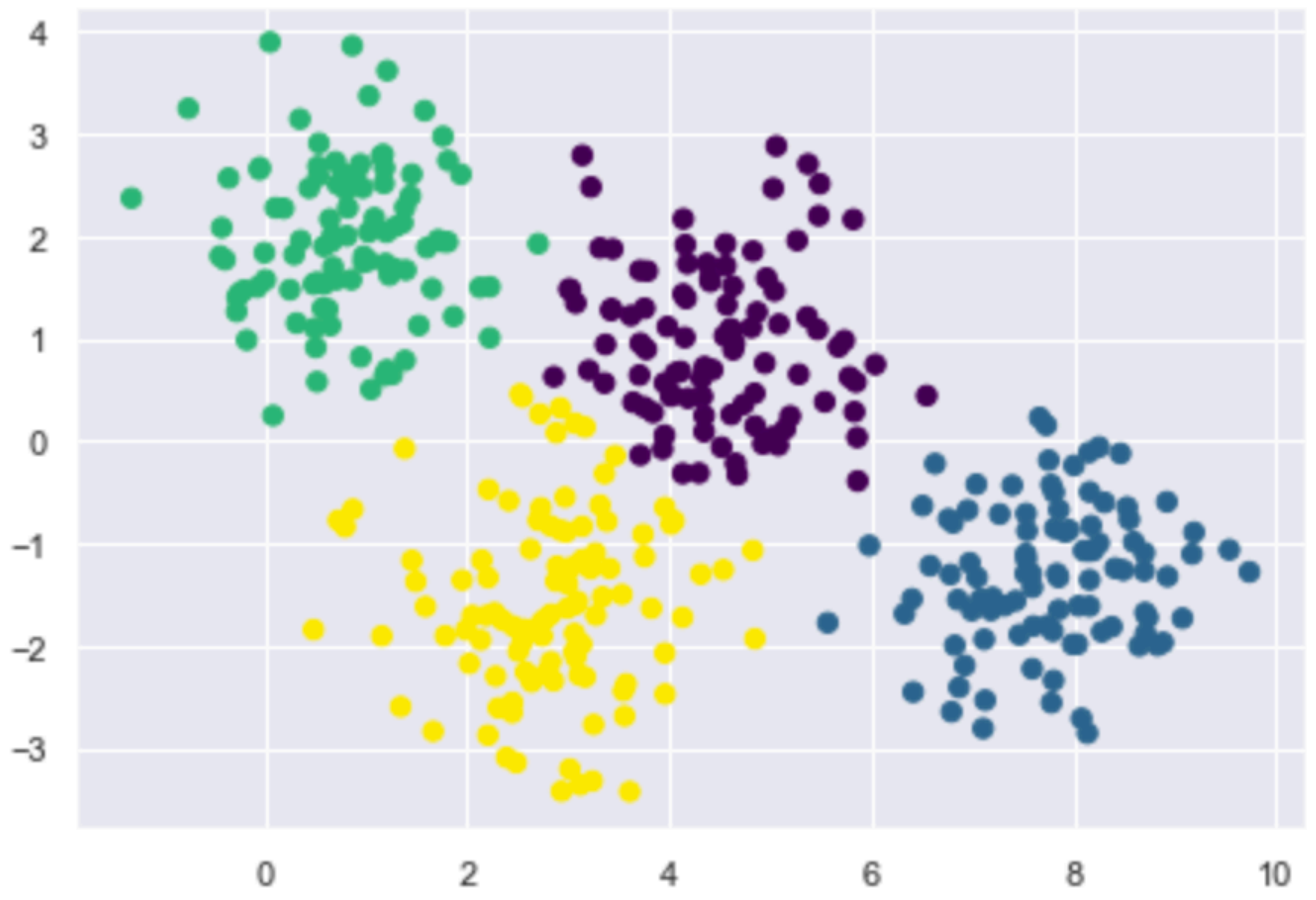
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu^{MLE}_{soft}, \Sigma^{MLE}_{soft})$$

$$\mu^{MLE}_{soft} = \frac{\sum_i p(t=2 \mid x, \theta)\, x_i}{\sum_i p(t=2 \mid x, \theta)}$$

$$\Sigma^{MLE}_{soft} = \frac{\sum_i p(t=2 \mid x_i, \theta)\,(x_i - \mu^{MLE}_{soft}) \times (x_i - \mu^{MLE}_{soft})^T}{\sum_i p(t=2 \mid x_i, \theta)}$$

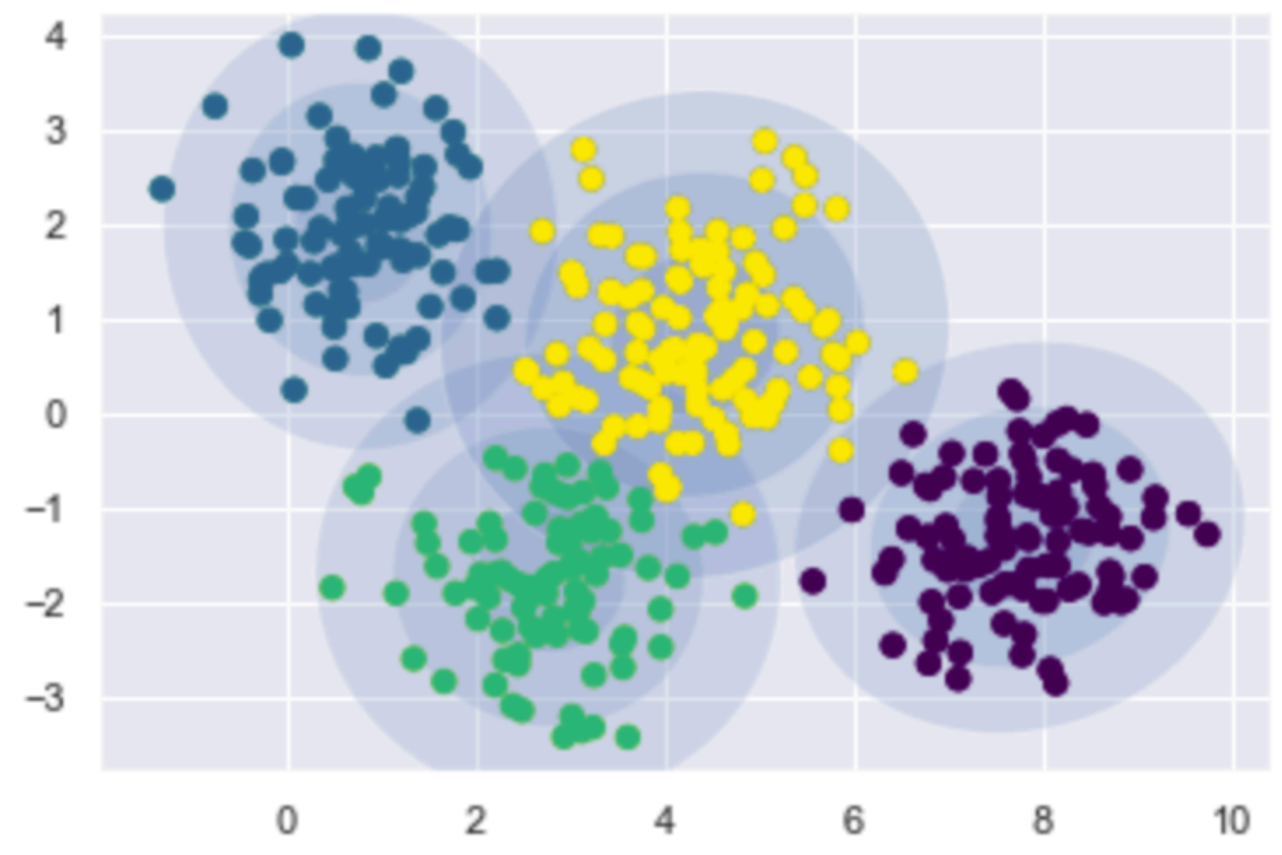
We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

STEP 5



2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [5/6]



Soft / probabilistic clustering : if we **know the source** of each instances then,

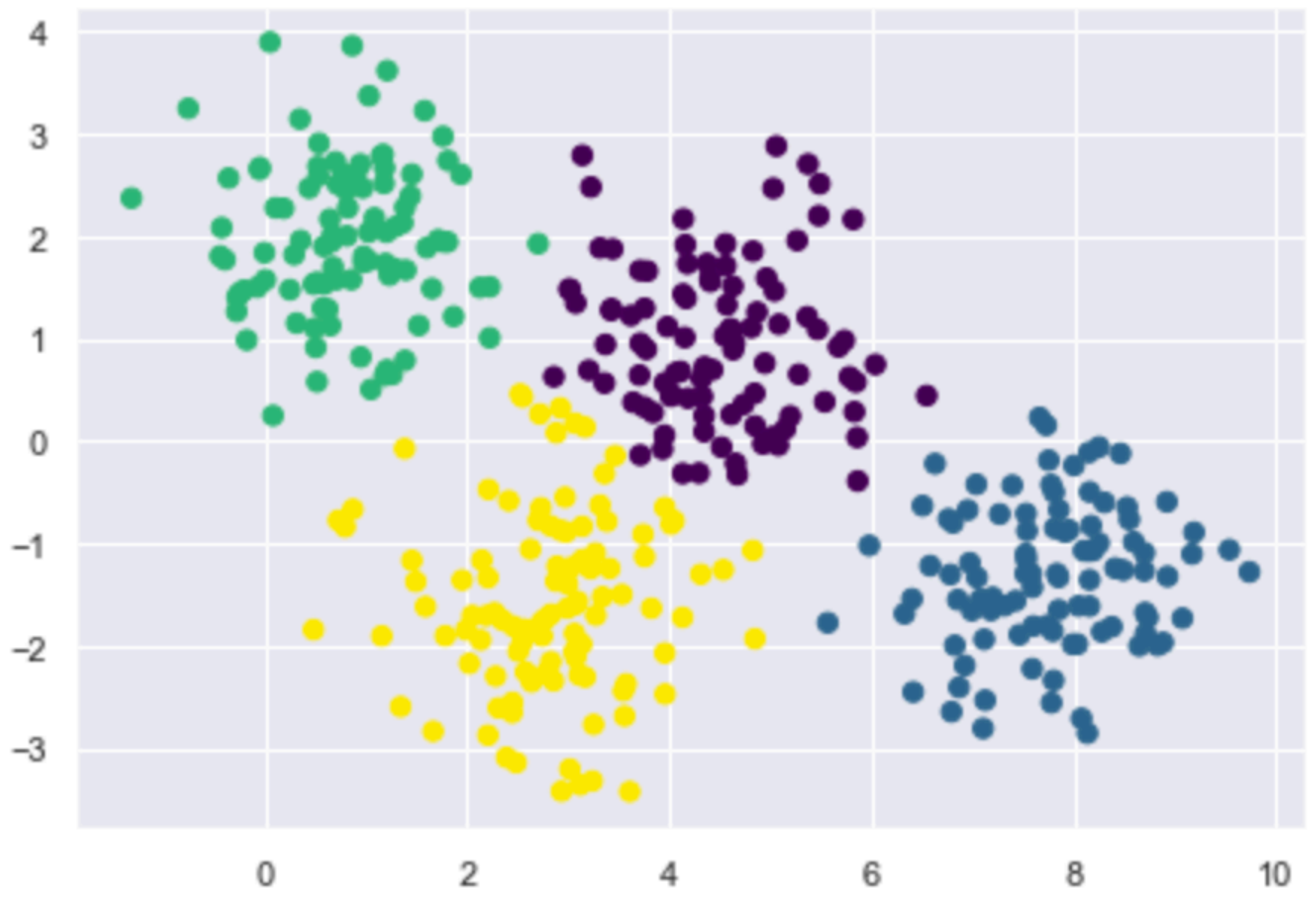
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

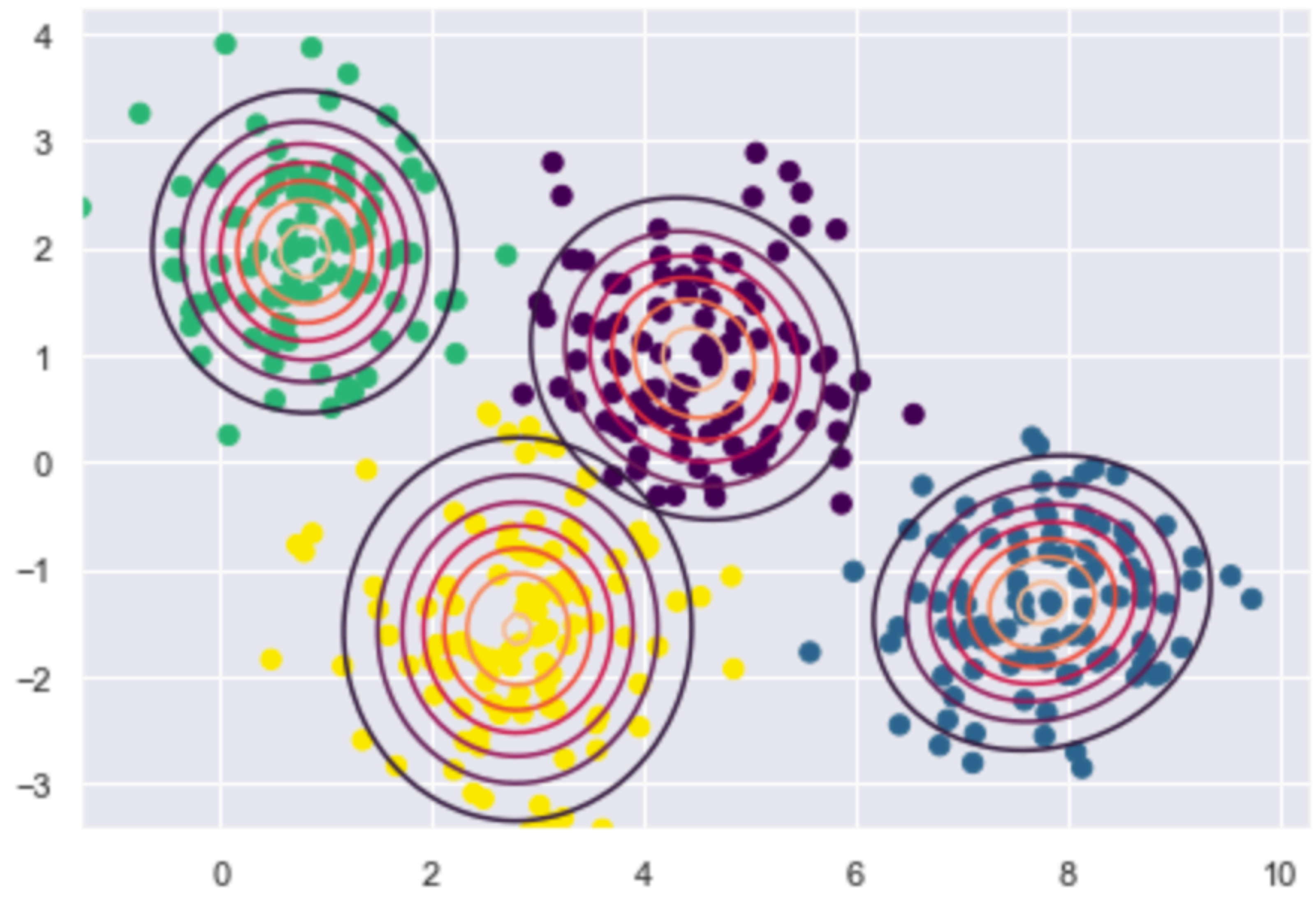
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

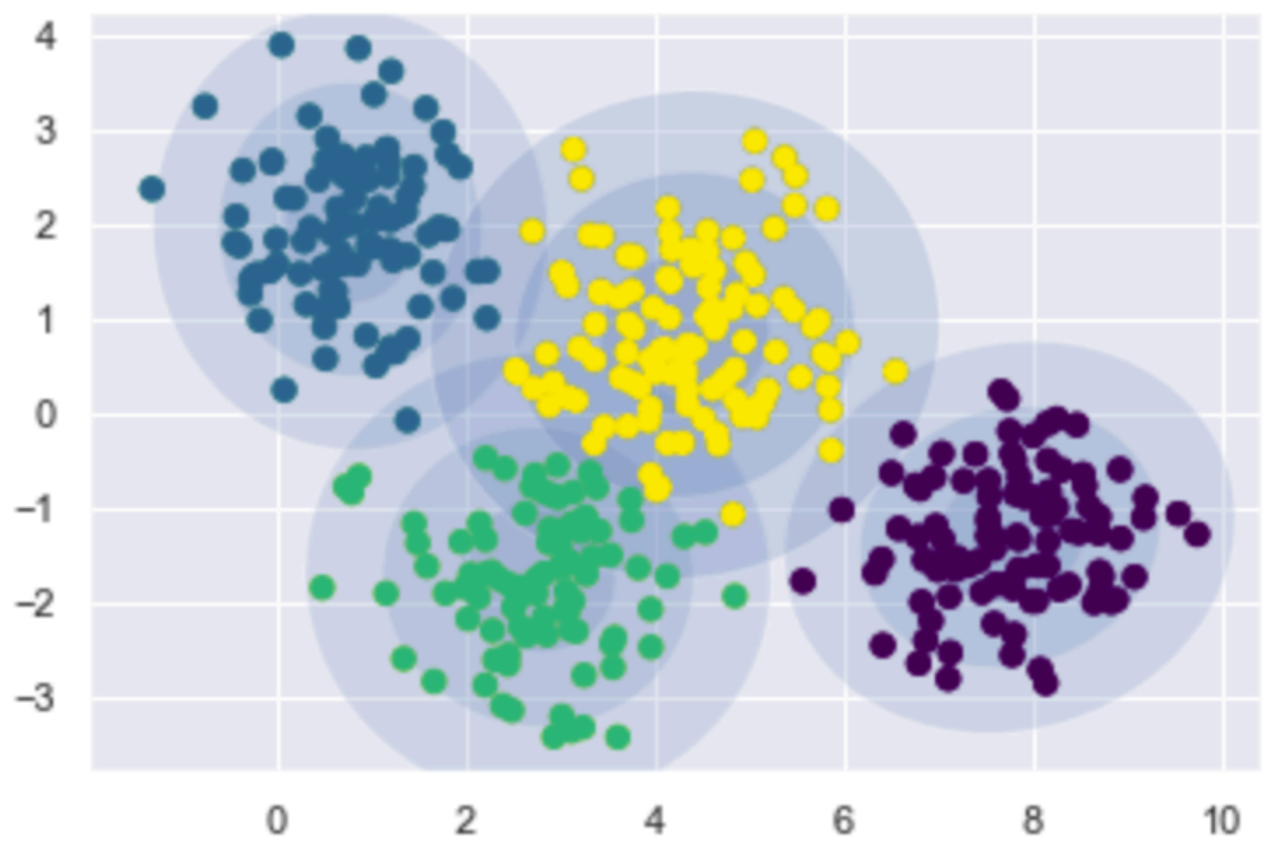
STEP 5





2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [6/6]



Soft / probabilistic clustering : **if we **know the source of each instances then,

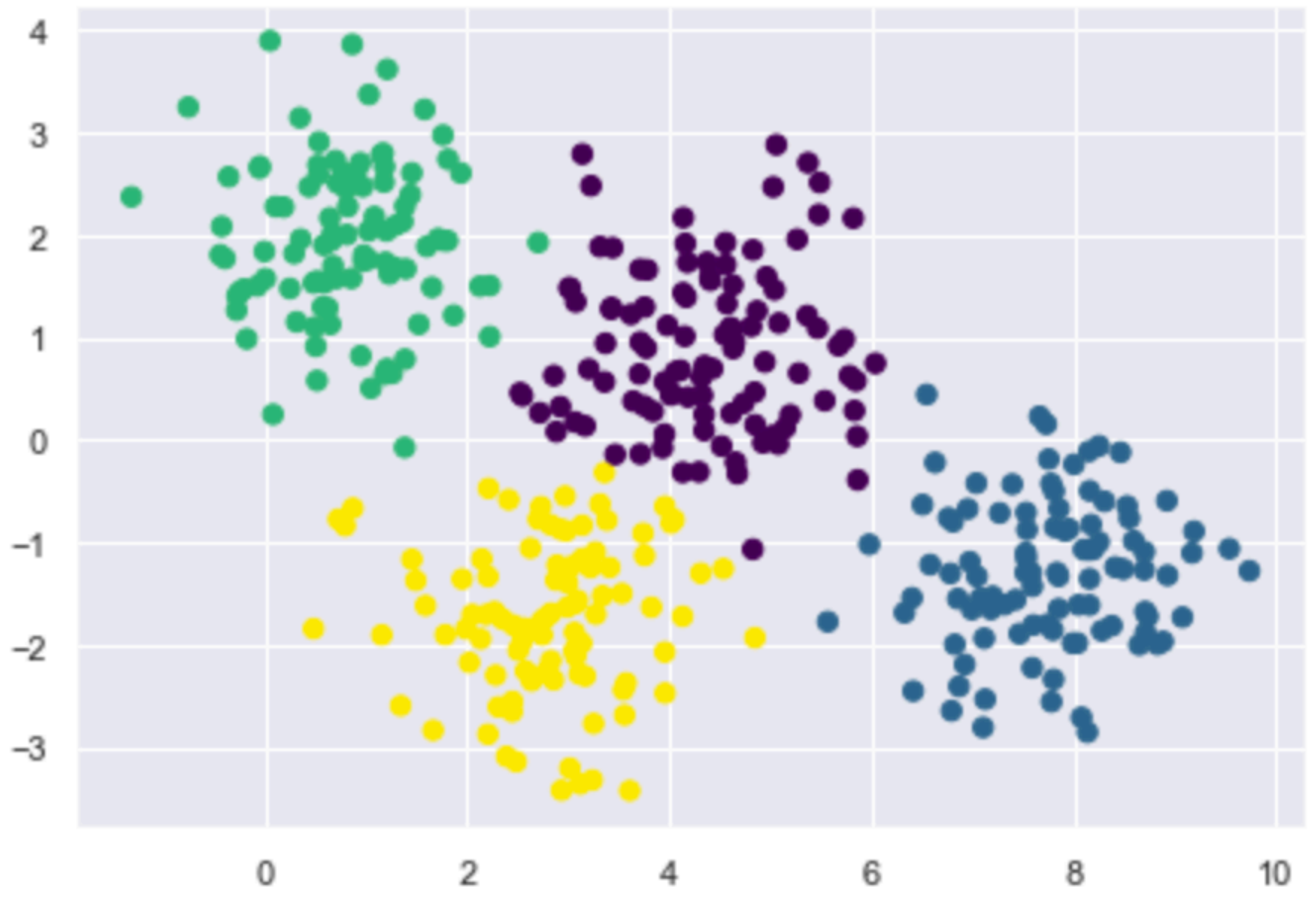
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

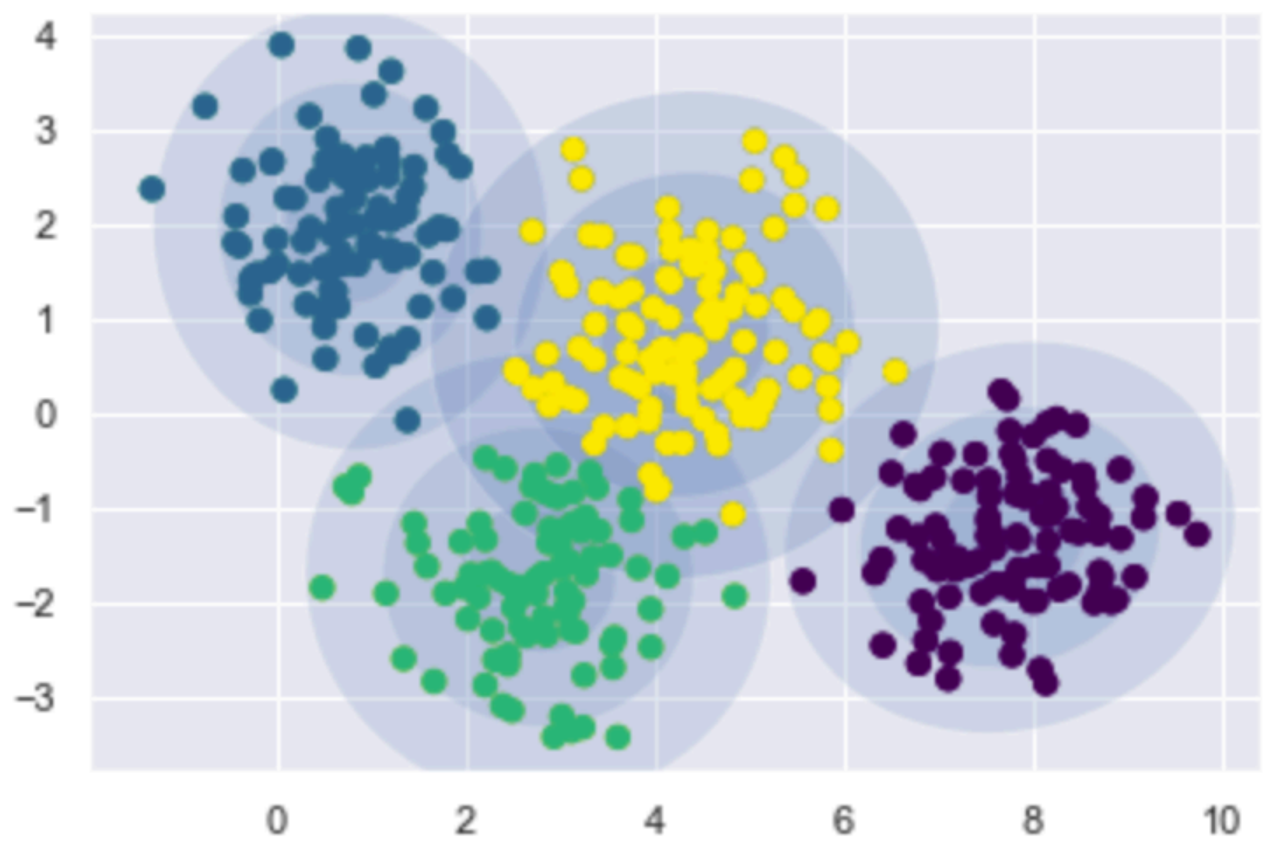
We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

STEP 6



2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [6/6]



Soft / probabilistic clustering : if we know the source of each instances then,

$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

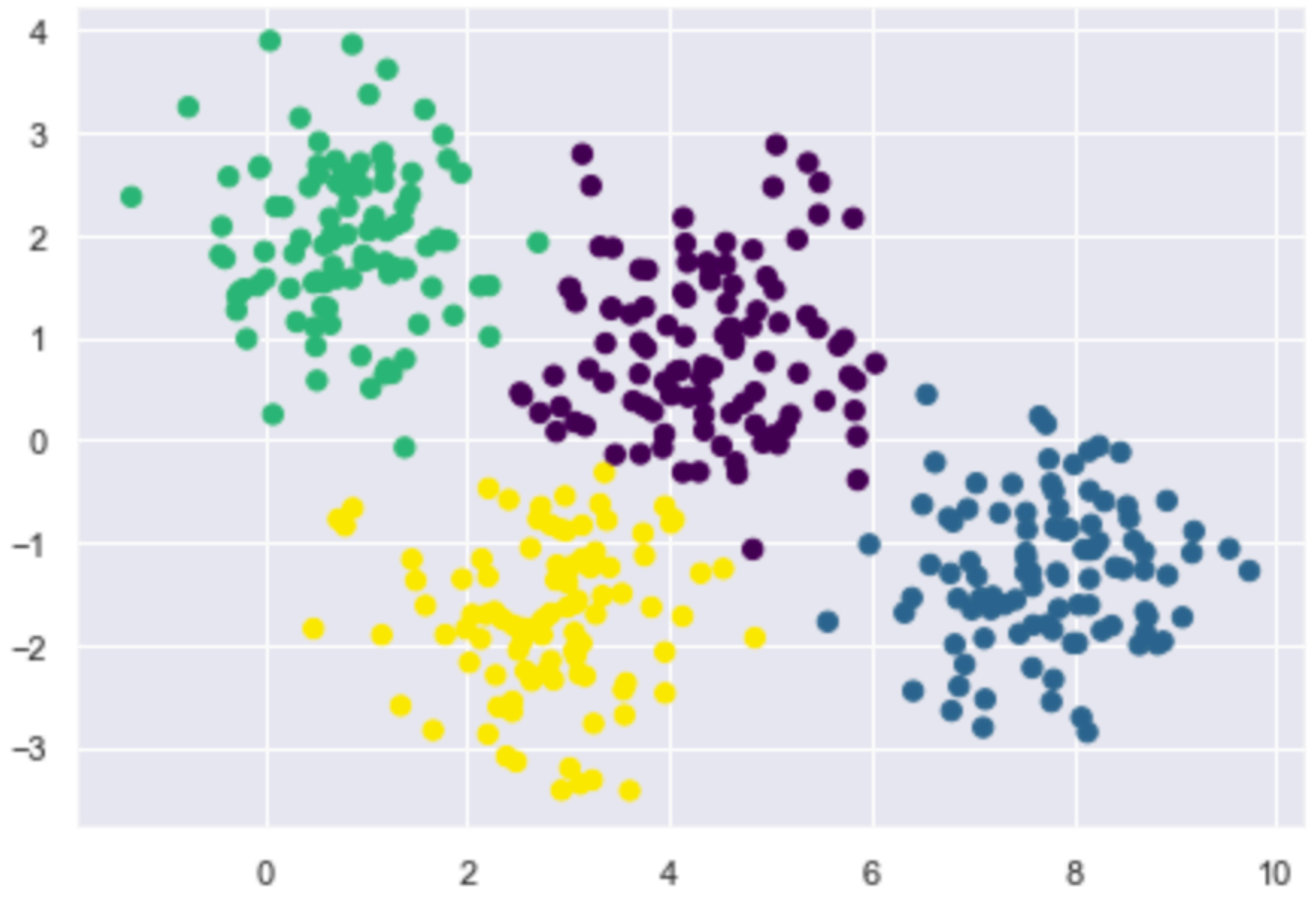
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

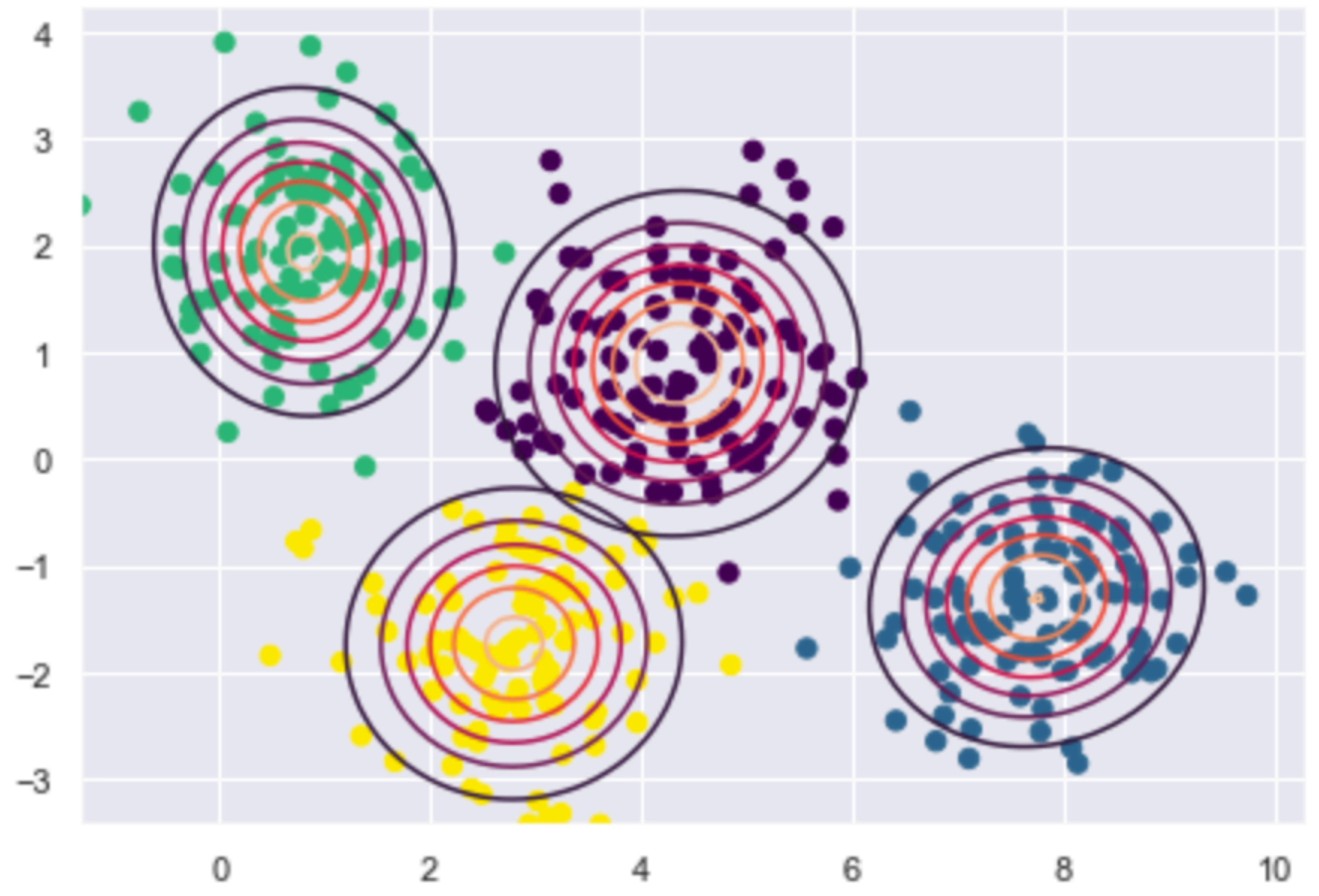
$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

We are now in the following situation :

- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

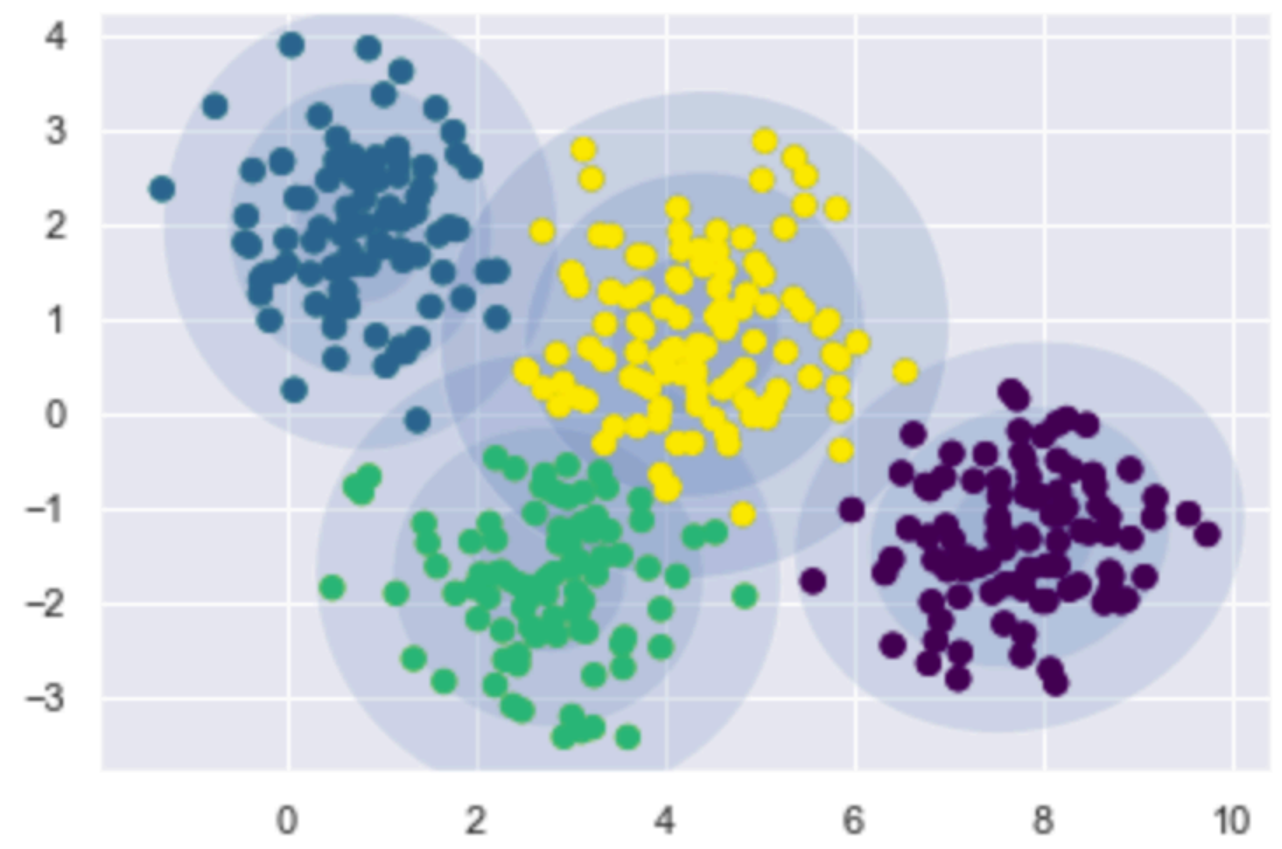
STEP 6





2. Probabilistic clustering

Gaussian Mixture Model : some intuitions for training this model [6/6]



Soft / probabilistic clustering : if we know the source of each instances then,

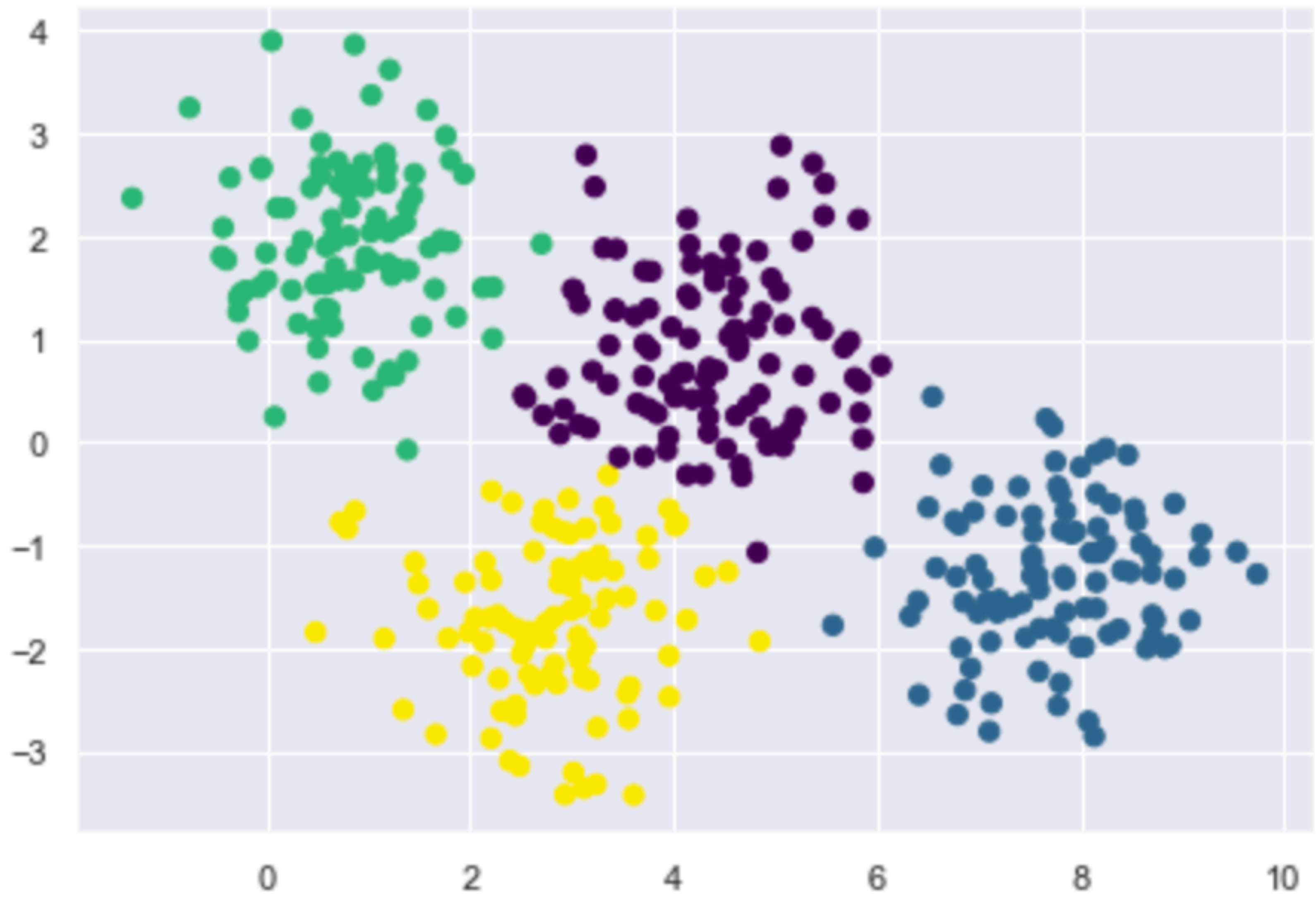
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

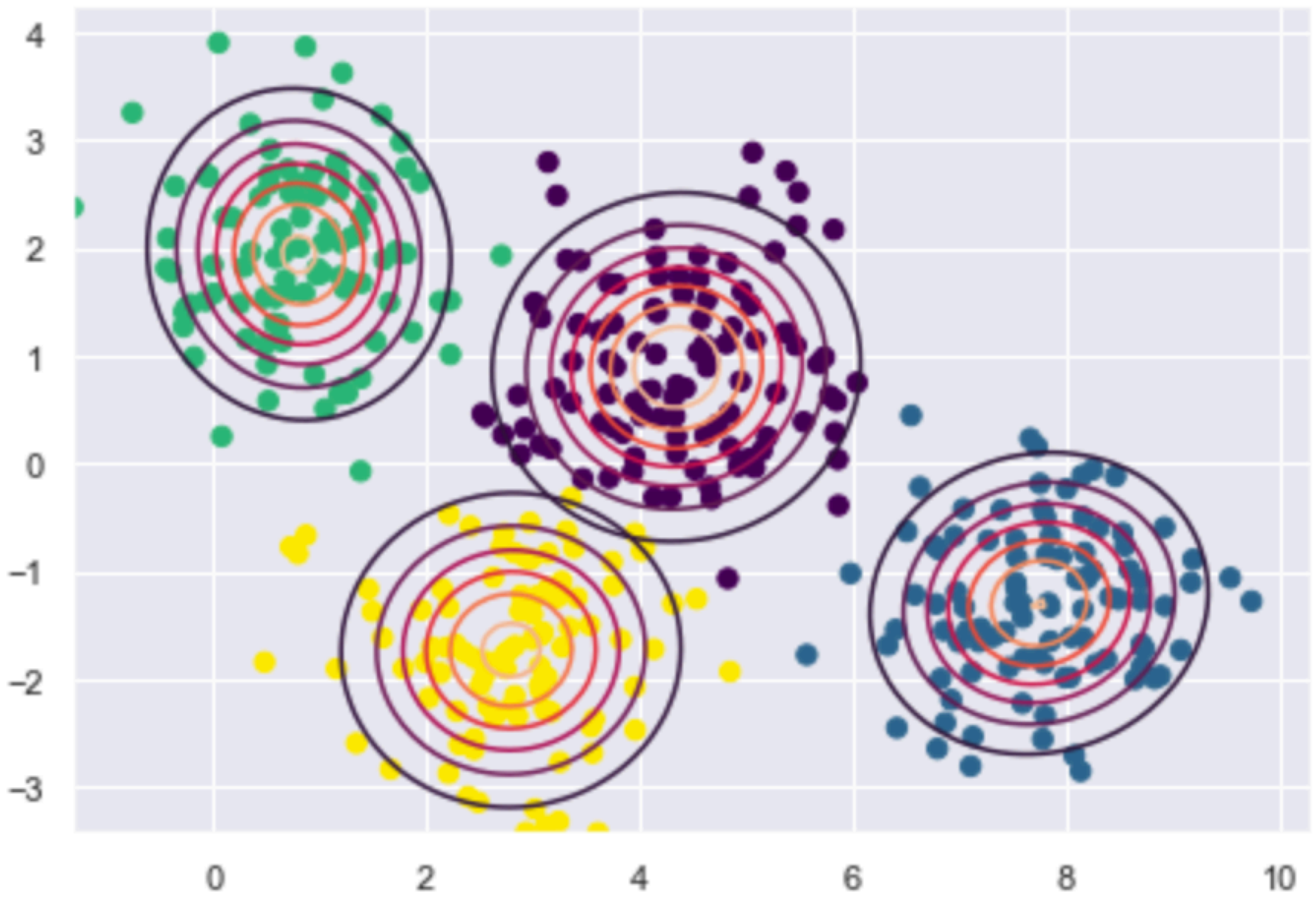
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\, (x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

We are now in the following situation :
- If we **knew the parameters**, we could compute the posteriors **(ESTIMATION)**
- If we **knew the posteriors**, we could easily compute the parameters **(MAXIMIZATION)**

STEP 6





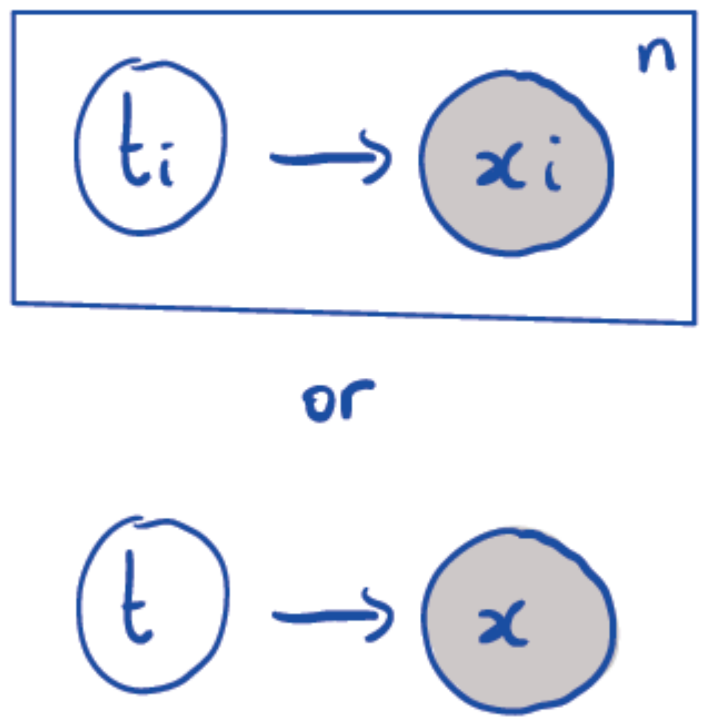
- **flexible probabilistic approach to clustering problem**

2.b EM-algorithm

2.b. Expectation-Maximization algorithm

Reminder : Maximum Likelihood Estimation (MLE)

Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \mid \theta) = \arg\max_{\theta} \log p(\mathbf{x} \mid \theta)$



$$p(x_i \mid \theta) = \sum_{k=1}^{4} p(x_i, t_i = k \mid \theta)$$

$$\hat{\theta} = \arg\max_{\theta} \left\{ \log P(X \mid \theta) \right\}$$

$$\log P(X \mid \theta) = \log \prod_{i=1}^{n} p(x_i \mid \theta) = \sum_{i=1}^{n} \log p(x_i \mid \theta)$$

$$= \sum_{i=1}^{n} \log \sum_{k=1}^{4} p(x_i, t_i = k \mid \theta)$$

$$= \sum_{i=1}^{n} \log \sum_{k=1}^{4} \frac{q(t_i = k)}{q(t_i = k)} p(x_i, t_i = k \mid \theta) \quad \text{for any distribution } q$$

$$\underset{\text{Jensen}}{\geq} \sum_{i=1}^{n} \sum_{k=1}^{4} q(t_i = k) \log \frac{p(x_i, t_i = k \mid \theta)}{q(t_i = k)} \quad \text{for any } q$$

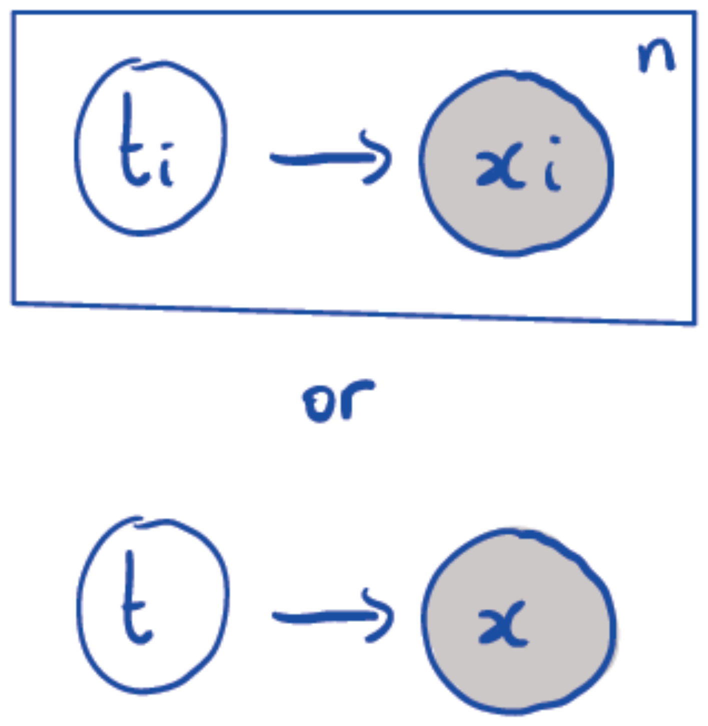
$$= \mathcal{L}(\theta, q) \quad \text{for any } \theta \text{ and } q$$

2.b. Expectation-Maximization algorithm

variational lower bound

Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$

$$\log P(X|\theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$$



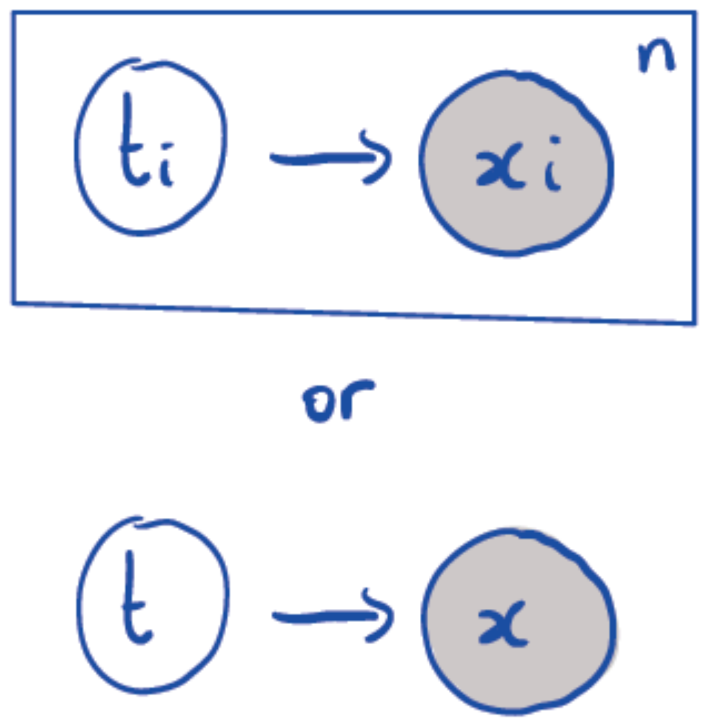
$$p(x_i|\theta) = \sum_{k=1}^{4} p(x_i, t_i = k|\theta)$$

$$\hat{\theta} = \arg\max_{\theta} \left\{ \log P(X|\theta) \right\}$$

2.b. Expectation-Maximization algorithm

variational lower bound

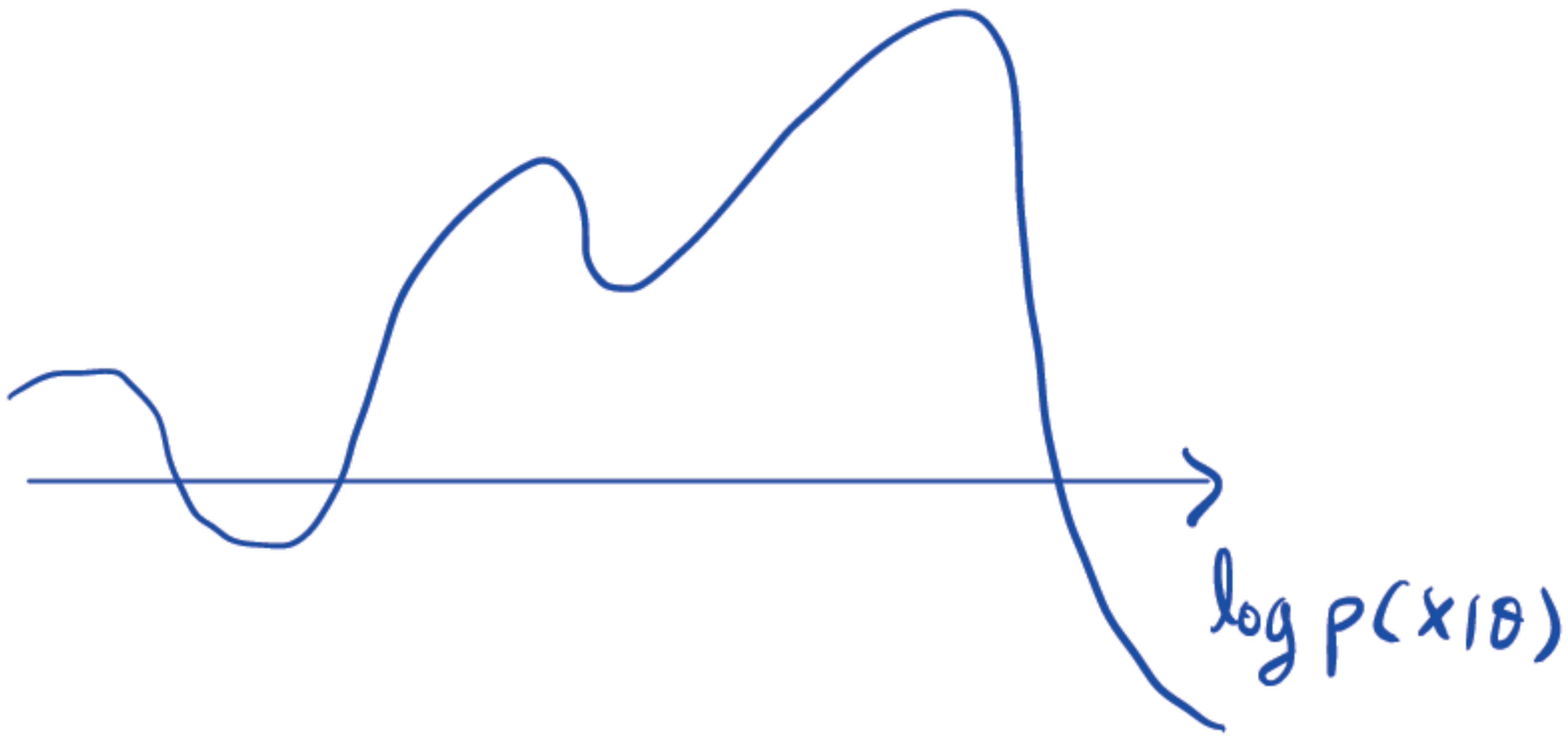
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \,|\, \theta) = \arg\max_{\theta} \log p(\mathbf{x} \,|\, \theta)$



$$p(x_i \,|\, \theta) = \sum_{k=1}^{4} p(x_i, t_i = k \,|\, \theta)$$

$$\hat{\theta} = \arg\max_{\theta} \left\{ \log P(X \,|\, \theta) \right\}$$

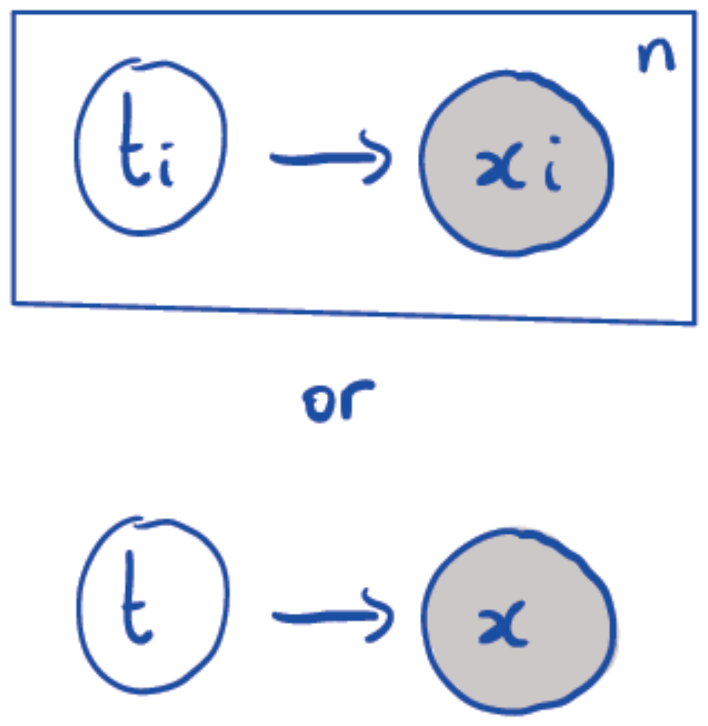
$$\log P(X \,|\, \theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$$



2.b. Expectation-Maximization algorithm

EM algorithm : E-step

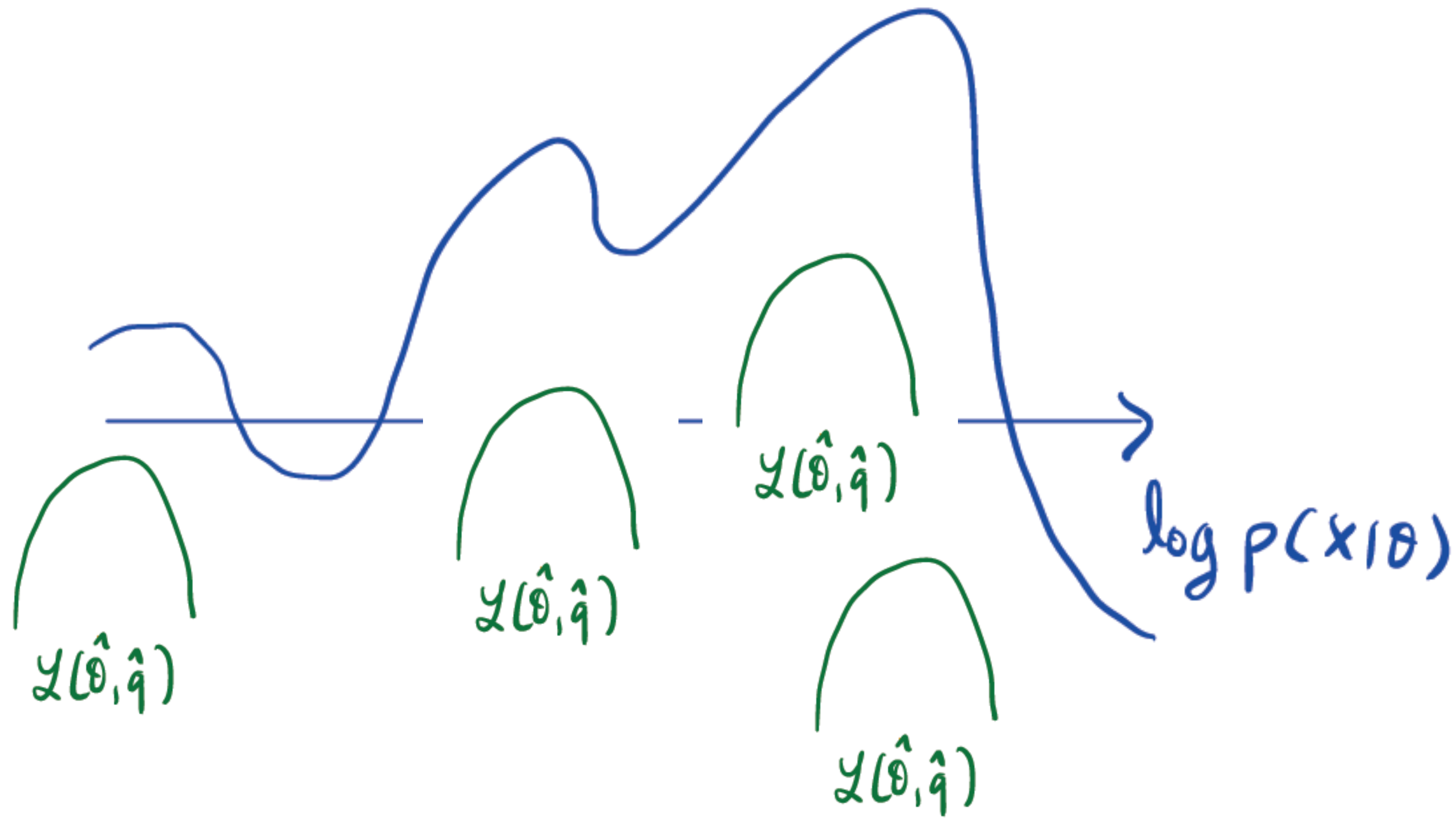
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$



$p(x_i|\theta) = \sum_{k=1}^{4} p(x_i, t_i = k|\theta)$

$\hat{\theta} = \arg\max_{\theta} \{ \log P(X|\theta) \}$

$\log P(X|\theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$



2.b. Expectation-Maximization algorithm

EM algorithm : E-step

Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$

$\boxed{t_i \longrightarrow x_i}^{\,n}$

or

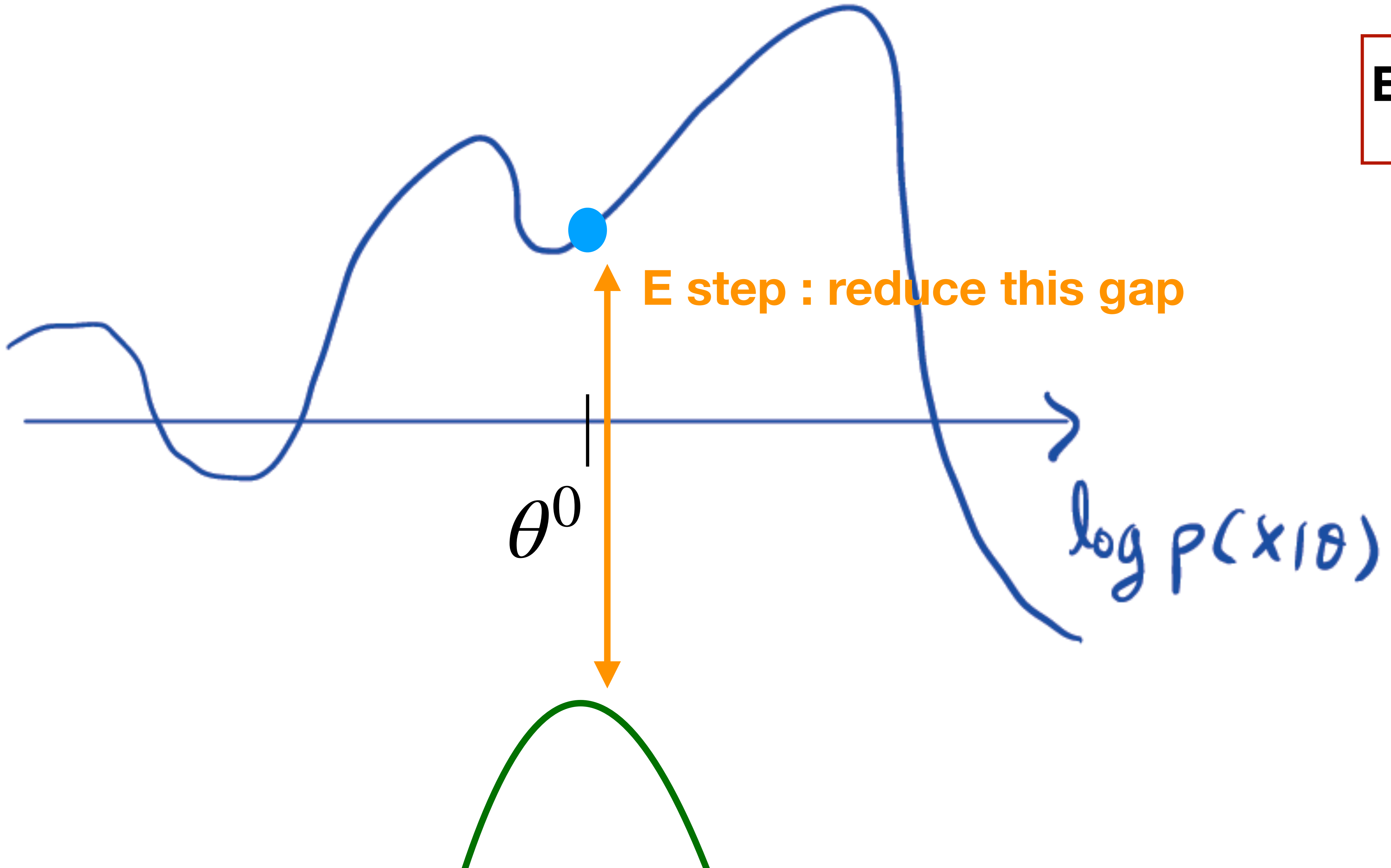
$t \longrightarrow x$

$p(x_i|\theta) = \sum_{k=1}^{4} p(x_i, t_i = k|\theta)$

$\hat{\theta} = \arg\max_{\theta} \{ \log P(X|\theta) \}$

$\log P(X|\theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$

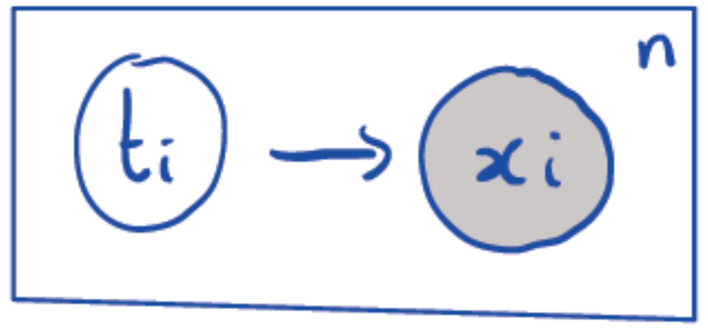
Expectation step : $q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)$



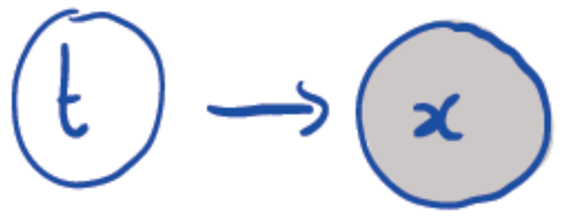
2.b. Expectation-Maximization algorithm

EM algorithm : E-step

Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$



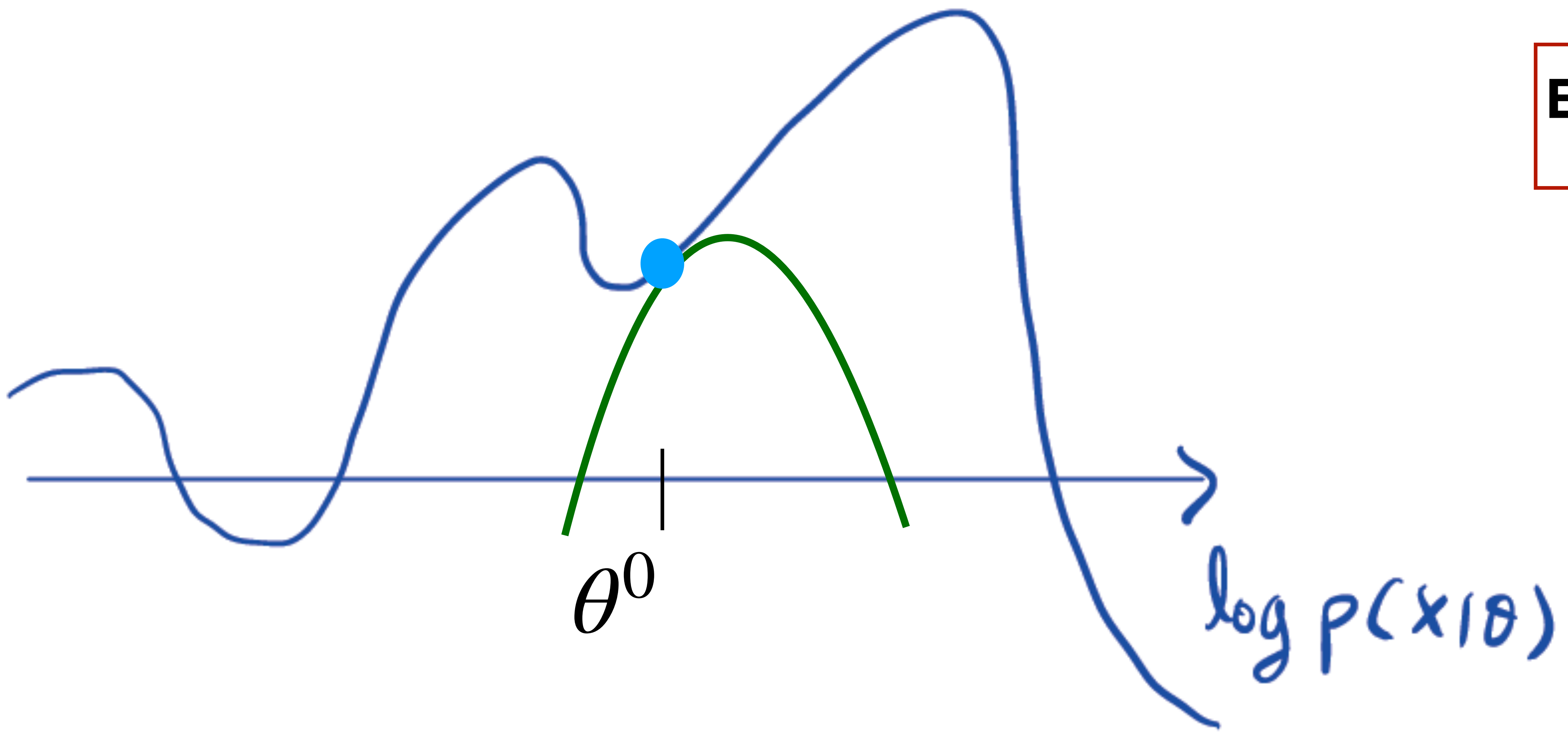
or



$$p(x_i|\theta) = \sum_{k=1}^{4} p(x_i, t_i = k|\theta)$$

$$\hat{\theta} = \arg\max_{\theta} \{ \log P(X|\theta) \}$$

$$\log P(X|\theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$$

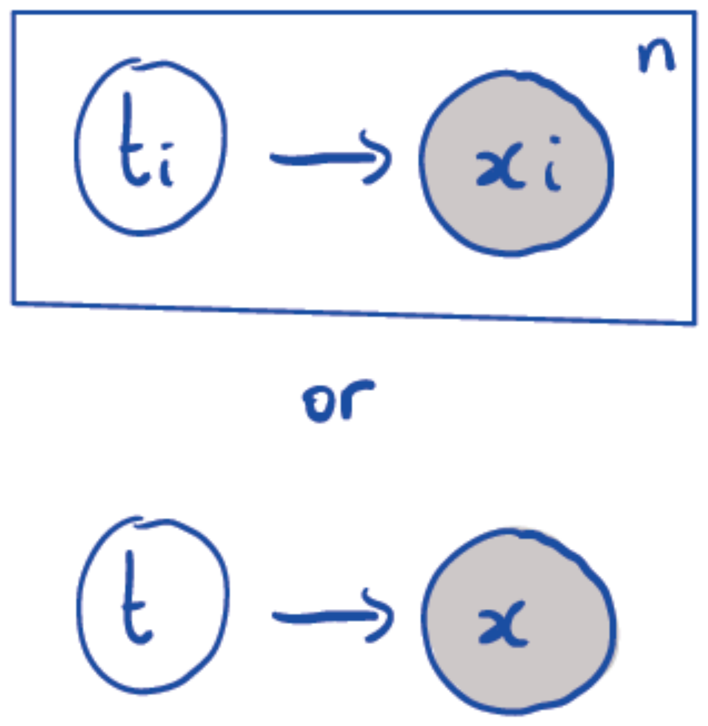


Expectation step : $q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)$

2.b. Expectation-Maximization algorithm

EM algorithm : M-step

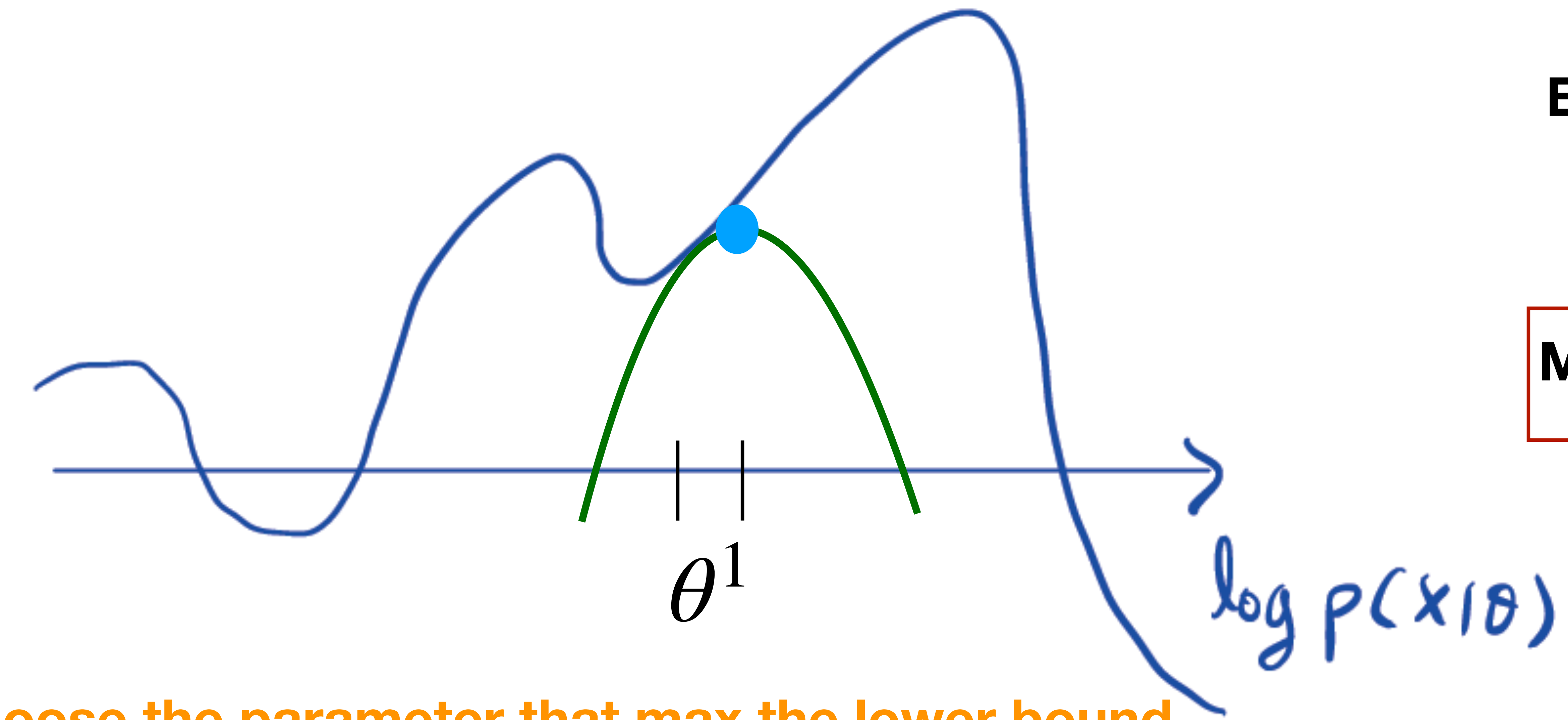
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$



$p(x_i|\theta) = \sum_{k=1}^{4} p(x_i, t_i = k|\theta)$

$\log P(X|\theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$

$\hat{\theta} = \arg\max_{\theta} \{ \log P(X|\theta) \}$



Expectation step : $q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)$

$\boxed{\textbf{Maximization step : } \theta^{k+1} = \arg\max_{\theta} \mathscr{L}(\theta, q^{k+1})}$

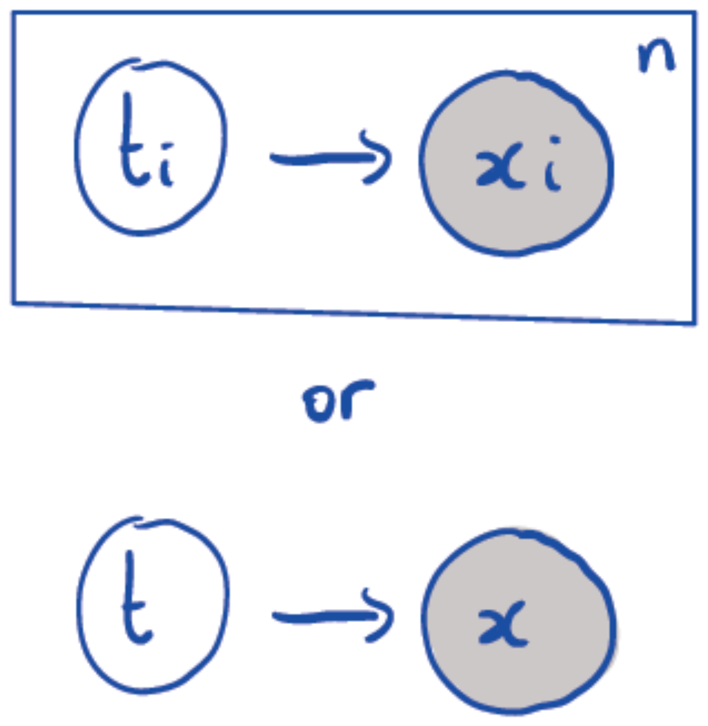
M step : choose the parameter that max the lower bound

2.b. Expectation-Maximization algorithm

EM algorithm

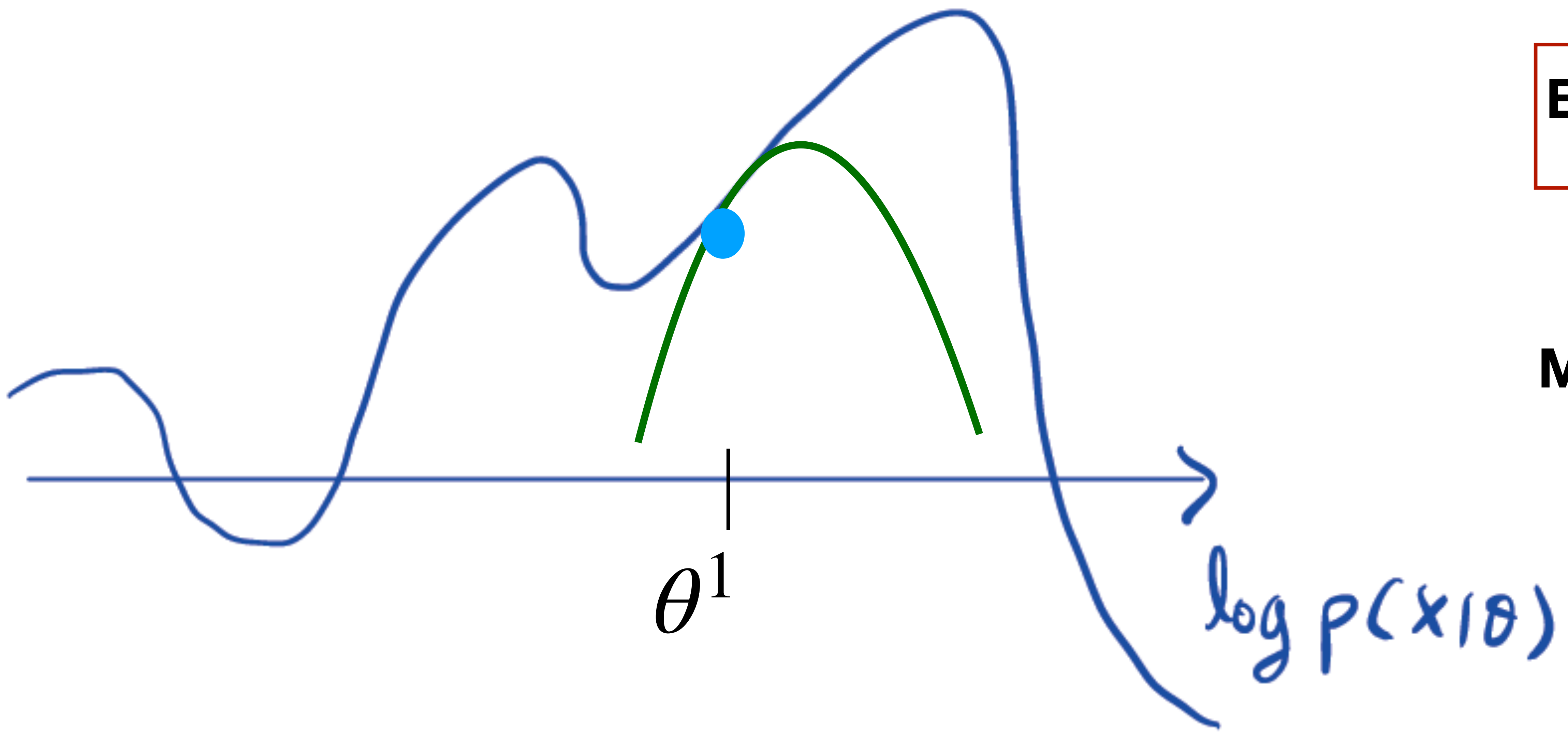
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \mid \theta) = \arg\max_{\theta} \log p(\mathbf{x} \mid \theta)$

$$\log P(x \mid \theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$$



$$p(x_i \mid \theta) = \sum_{k=1}^{4} p(x_i, t_i = k \mid \theta)$$

$$\hat{\theta} = \arg\max_{\theta} \{ \log P(x \mid \theta) \}$$



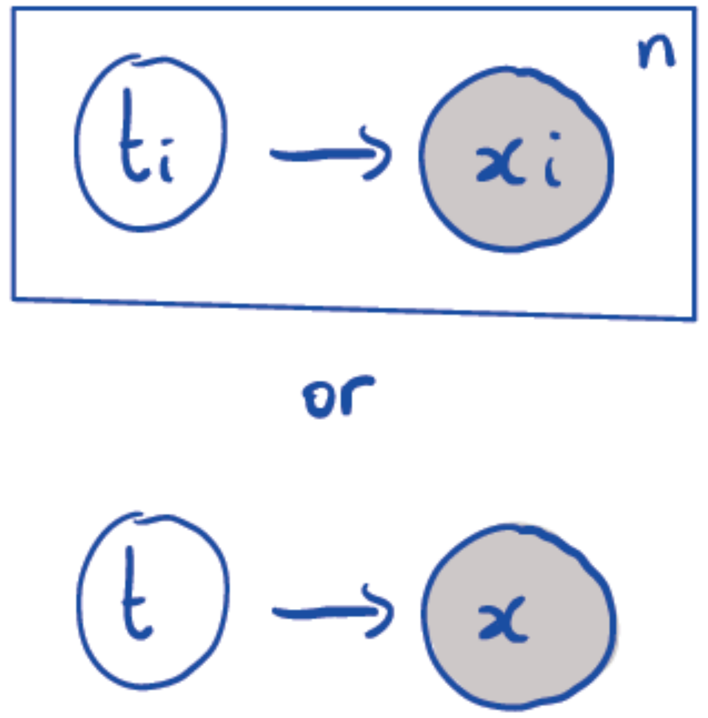
$$\boxed{\textbf{Expectation step : } q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)}$$

Maximization step : $\theta^{k+1} = \arg\max_{\theta} \mathscr{L}(\theta, q^{k+1})$

2.b. Expectation-Maximization algorithm

EM algorithm

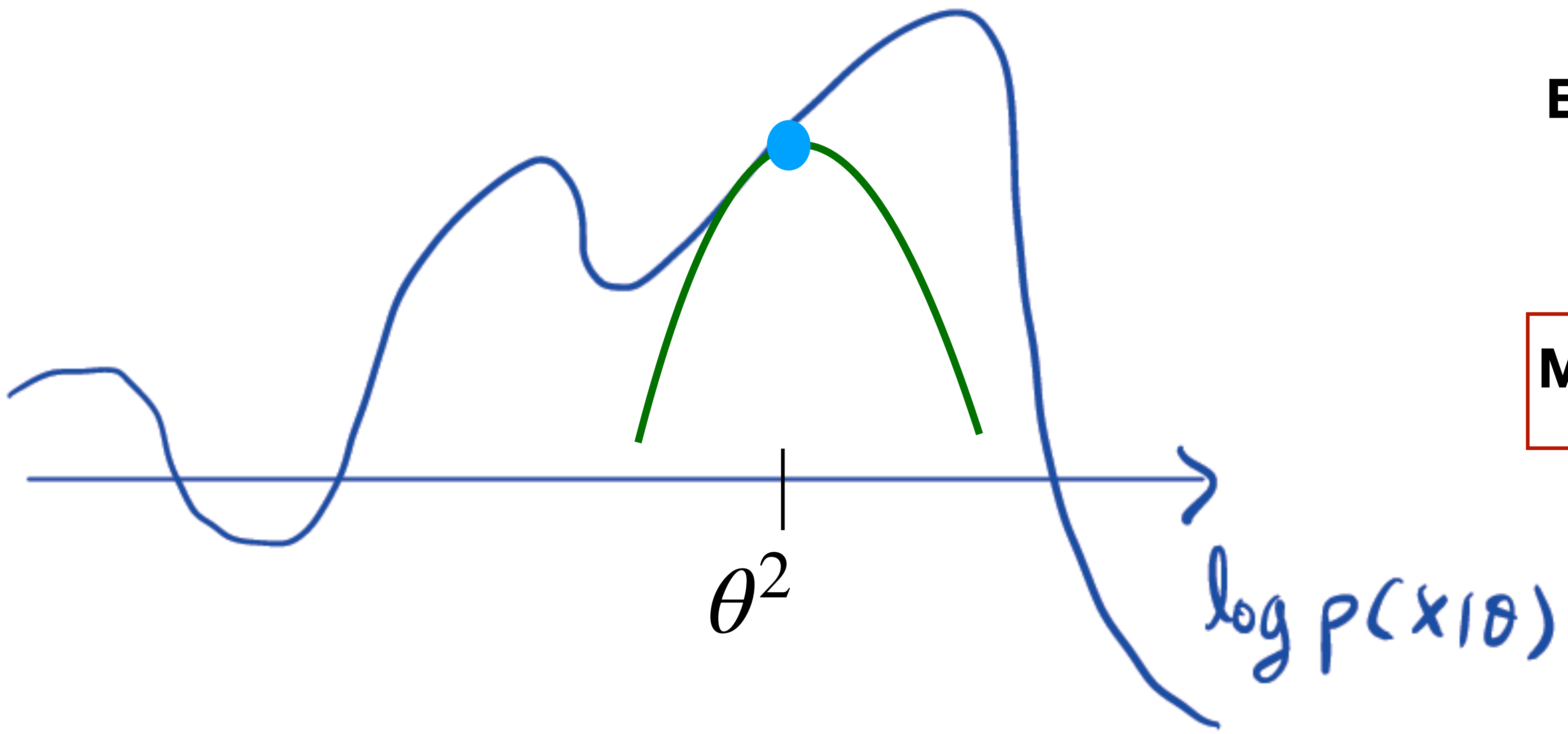
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \mid \theta) = \arg\max_{\theta} \log p(\mathbf{x} \mid \theta)$



$p(x_i \mid \theta) = \sum_{k=1}^{4} p(x_i, t_i = k \mid \theta)$

$\log P(X \mid \theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$

$\hat{\theta} = \arg\max_{\theta} \{ \log P(X \mid \theta) \}$



Expectation step : $q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)$

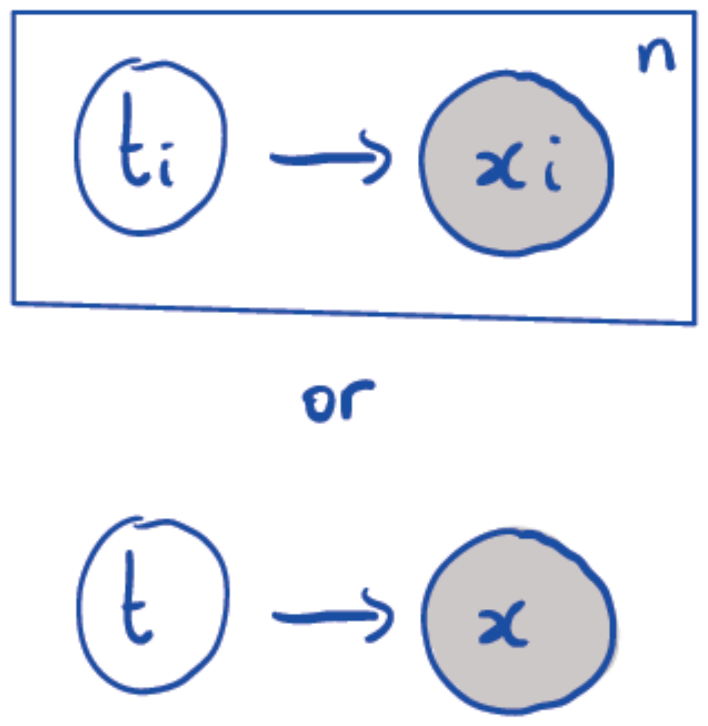
$\boxed{\textbf{Maximization step : } \theta^{k+1} = \arg\max_{\theta} \mathscr{L}(\theta, q^{k+1})}$

2.b. Expectation-Maximization algorithm

EM algorithm

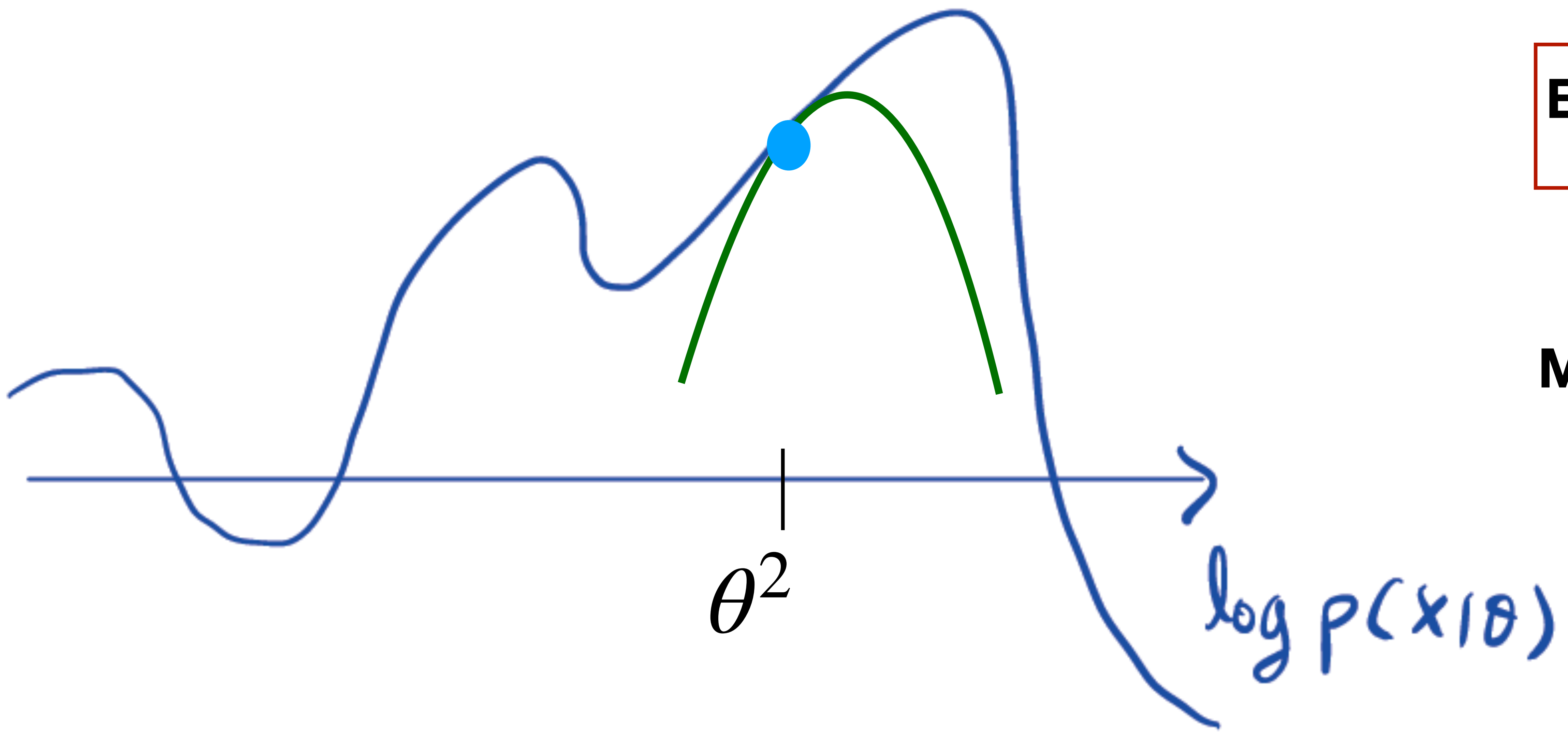
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$

$$\log P(X|\theta) \underset{\text{Jensen}}{\geq} \mathscr{L}(\theta, q)$$



$$p(x_i|\theta) = \sum_{k=1}^{4} p(x_i, t_i = k|\theta)$$

$$\hat{\theta} = \arg\max_{\theta} \{ \log P(X|\theta) \}$$



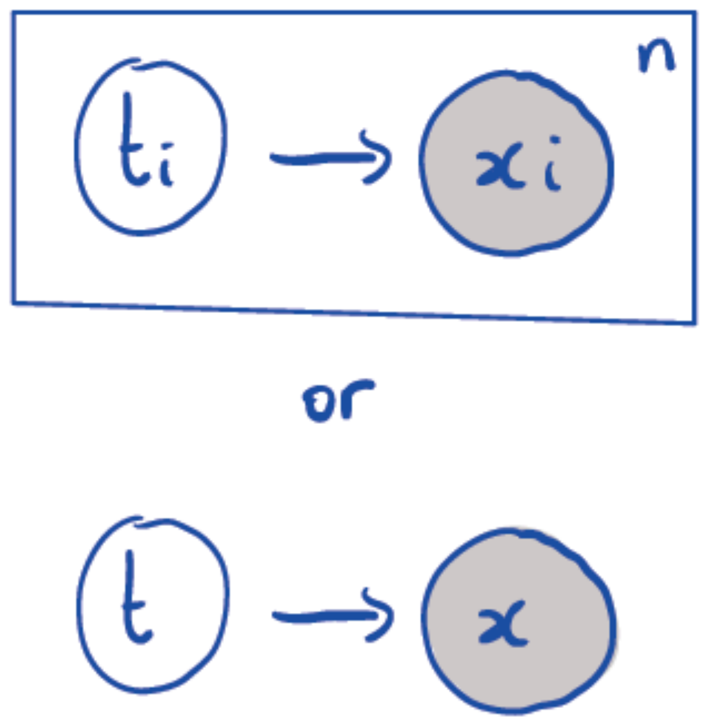
$\boxed{\textbf{Expectation step : } q^{k+1} = \arg\max_{q \in Family} \mathscr{L}(\theta^k, q)}$

Maximization step : $\theta^{k+1} = \arg\max_{\theta} \mathscr{L}(\theta, q^{k+1})$

2.b. Expectation-Maximization algorithm

EM algorithm

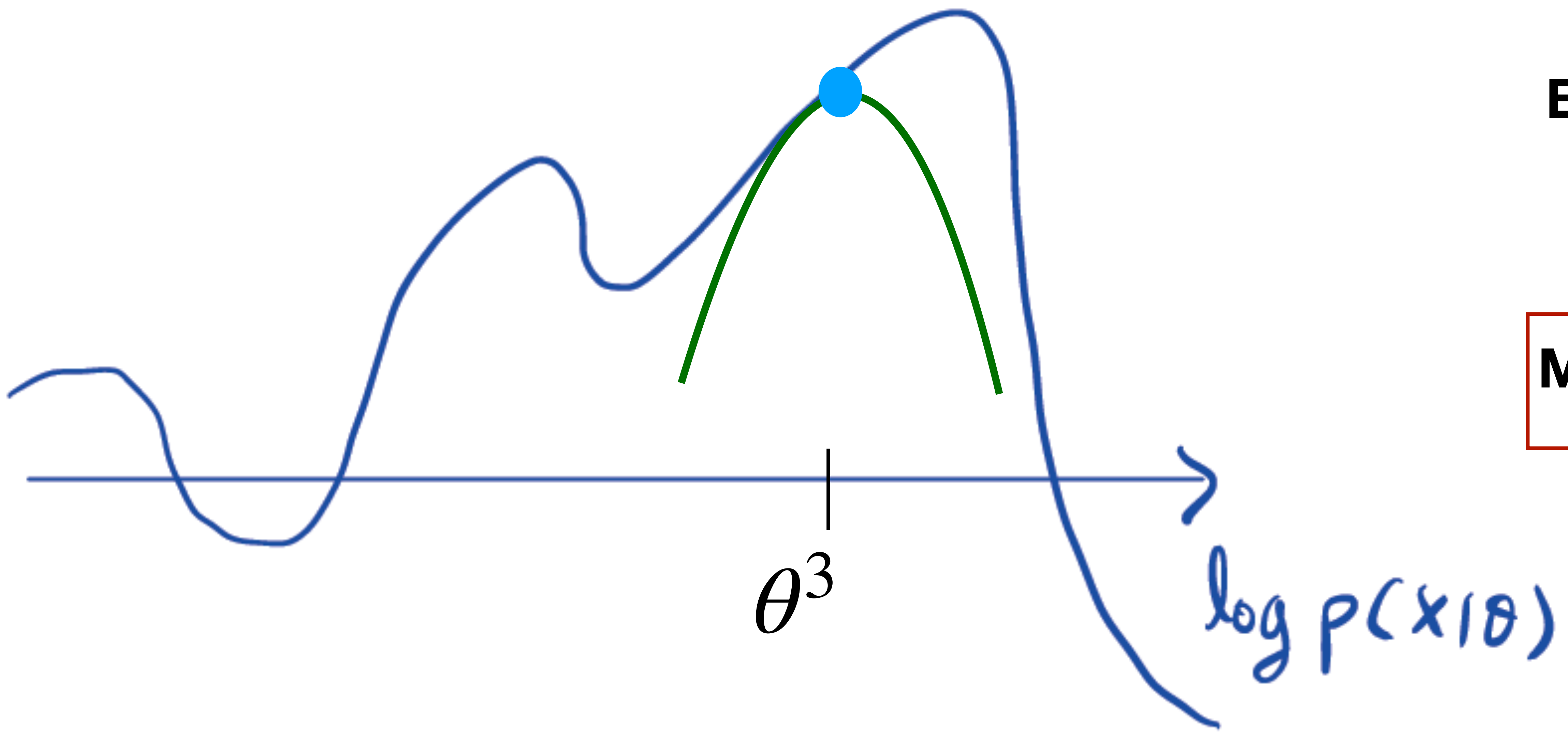
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$



$$p(x_i|\theta) = \sum_{k=1}^{4} p(x_i, t_i = k|\theta)$$

$$\log P(X|\theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$$

$$\hat{\theta} = \arg\max_{\theta} \left\{ \log P(X|\theta) \right\}$$



Expectation step : $q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)$

$$\boxed{\textbf{Maximization step : } \theta^{k+1} = \arg\max_{\theta} \mathscr{L}(\theta, q^{k+1})}$$

And so on … until we reach a local maximum

2.b. Expectation-Maximization algorithm

EM algorithm : more details

E-step :

$$q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q) \iff q(t_i) = p(t_i \,|\, x_i, \theta)$$

M-step :

$$\theta^{k+1} = \arg \max_{\theta} \mathscr{L}(\theta, q^{k+1}) \iff \theta^{k+1} = \arg \max_{\theta} \mathbb{E}_{q^{k+1}}[\log p(X, T \,|\, \theta)]$$

2.b. Expectation-Maximization algorithm

EM algorithm : back to GMM

E-step :

$$q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q) \iff q(t_i) = p(t_i \mid x_i, \theta)$$

GMM : for each point we indeed computed $q(t_i) = p(t_i \mid x_i, \theta)$

M-step :

$$\theta^{k+1} = \arg \max_{\theta} \mathscr{L}(\theta, q^{k+1}) \iff \theta^{k+1} = \arg \max_{\theta} \mathbb{E}_{q^{k+1}}[\log p(X, T \mid \theta)]$$

GMM : we updated the gaussian parameters with

$$\mu_{soft}^{MLE} = \frac{\sum_i \; p(\; t = 2 \mid x, \theta) \; x_i}{\sum_i p(\; t = 2 \mid x, \theta)}$$

which indeed is the M-step of the EM algorithm

2.b. Expectation-Maximization algorithm

EM algorithm : back to GMM

E-step :

$$q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q) \iff q(t_i) = p(t_i \mid x_i, \theta)$$

GMM : for each point we indeed computed $q(t_i) = p(t_i \mid x_i, \theta)$

M-step :

$$\theta^{k+1} = \arg\max_{\theta} \mathscr{L}(\theta, q^{k+1}) \iff \theta^{k+1} = \arg\max_{\theta} \mathbb{E}_{q^{k+1}}[\log p(X, T \mid \theta)]$$

GMM : we updated the gaussian parameters with

$$\mu_{soft}^{MLE} = \frac{\sum_i \; p(\, t = 2 \mid x, \theta) \; x_i}{\sum_i p(\; t = 2 \mid x, \theta)}$$

which indeed is the M-step of the EM algorithm

$$\sum_{i=1}^{n} E_{q(t_i)} \log p(x_i, t_i \mid \theta) = \sum_{i=1}^{n} \sum_{k=1}^{4} q(t_i = k) \log \left(\frac{1}{const} e^{-\frac{(x_i - \mu_k)^2}{2\sigma_k^2}} \times \pi_k \right)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{4} q(t_i = k) \left(\log \left(\frac{\pi_k}{const} \right) - \frac{(x_i - \mu_k)^2}{2\sigma_k^2} \right)$$

$$\frac{\partial}{\partial \mu_2} \left(\sum_{i=1}^{n} \sum_{k=1}^{4} q(t_i = k) \left(\log \left(\frac{\pi_k}{const} \right) - \frac{(x_i - \mu_k)^2}{2\sigma_k^2} \right) \right)$$

$$= \sum_{i=1}^{n} q(t_i = 2) \left(0 + \frac{(x_i - \mu_2)}{\cancel{\sigma_2^2}} \right) = 0$$

$$\iff \sum_{i=1}^{n} q(t_i = 2) \times x_i - \mu_2 \sum_{i=1}^{n} q(t_i = 2) = 0$$

$$\iff \boxed{\mu_2 = \frac{\sum_{i=1}^{n} q(t_i = 2) \times x_i}{\sum_{i=1}^{n} q(t_i = 2)}}$$

3 Probabilistic dimensionality reduction and EM-algorithm

3. Probabilistic dimensionality reduction

Dimensionality reduction : reminder

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

3. Probabilistic dimensionality reduction

Dimensionality reduction : reminder

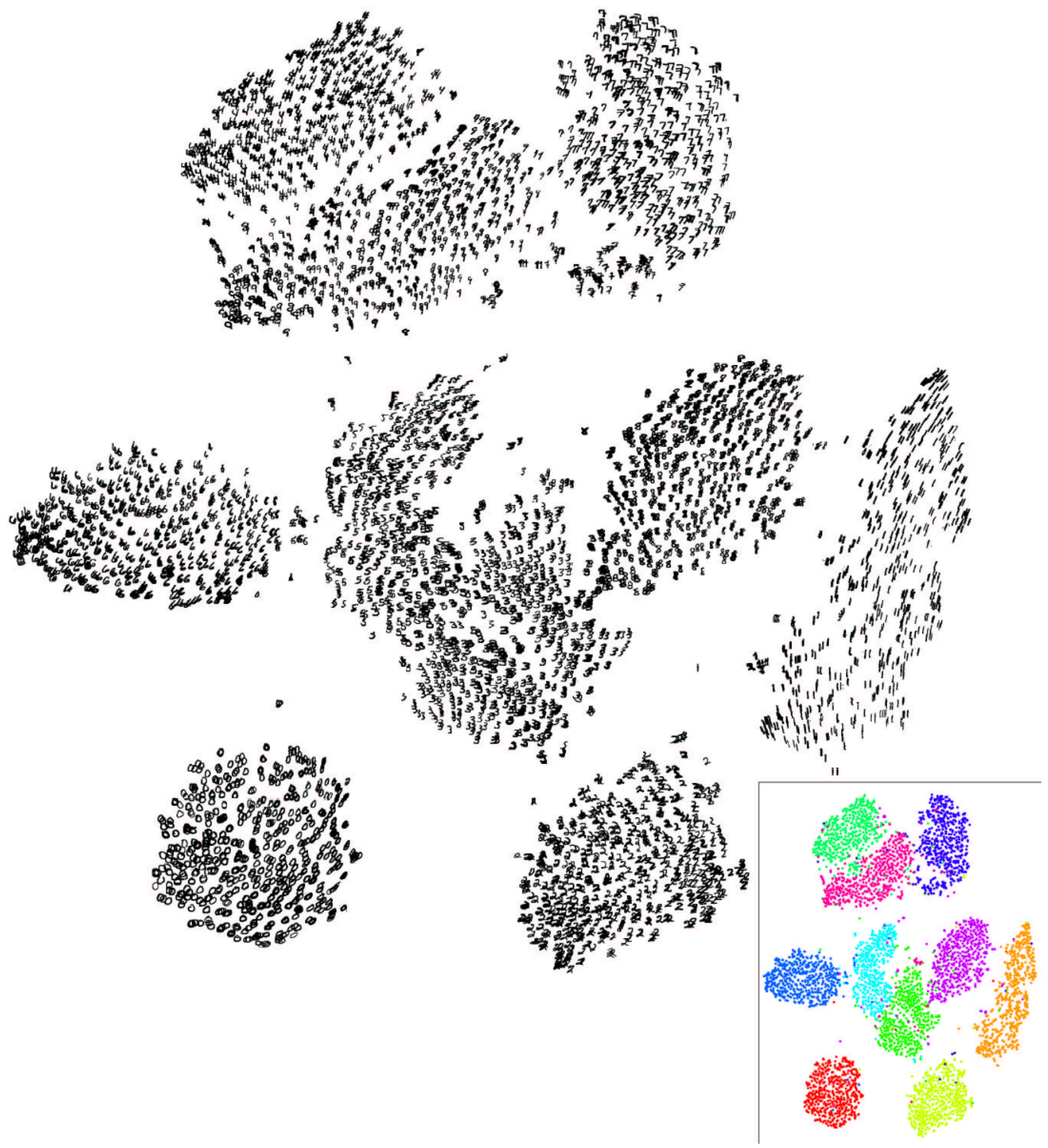
Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

Why do we care ?

- Avoid **curse of dimensionality** :
  a high-dimensional data can be dangerous if the data is too sparse

- **Noise reduction** :
  In a High-dimensional dataset there might be too much noise.

- **Data visualisation (2D or 3D visualisation)** :
  We cannot visualise a high-dimensional data (dimension > 3)



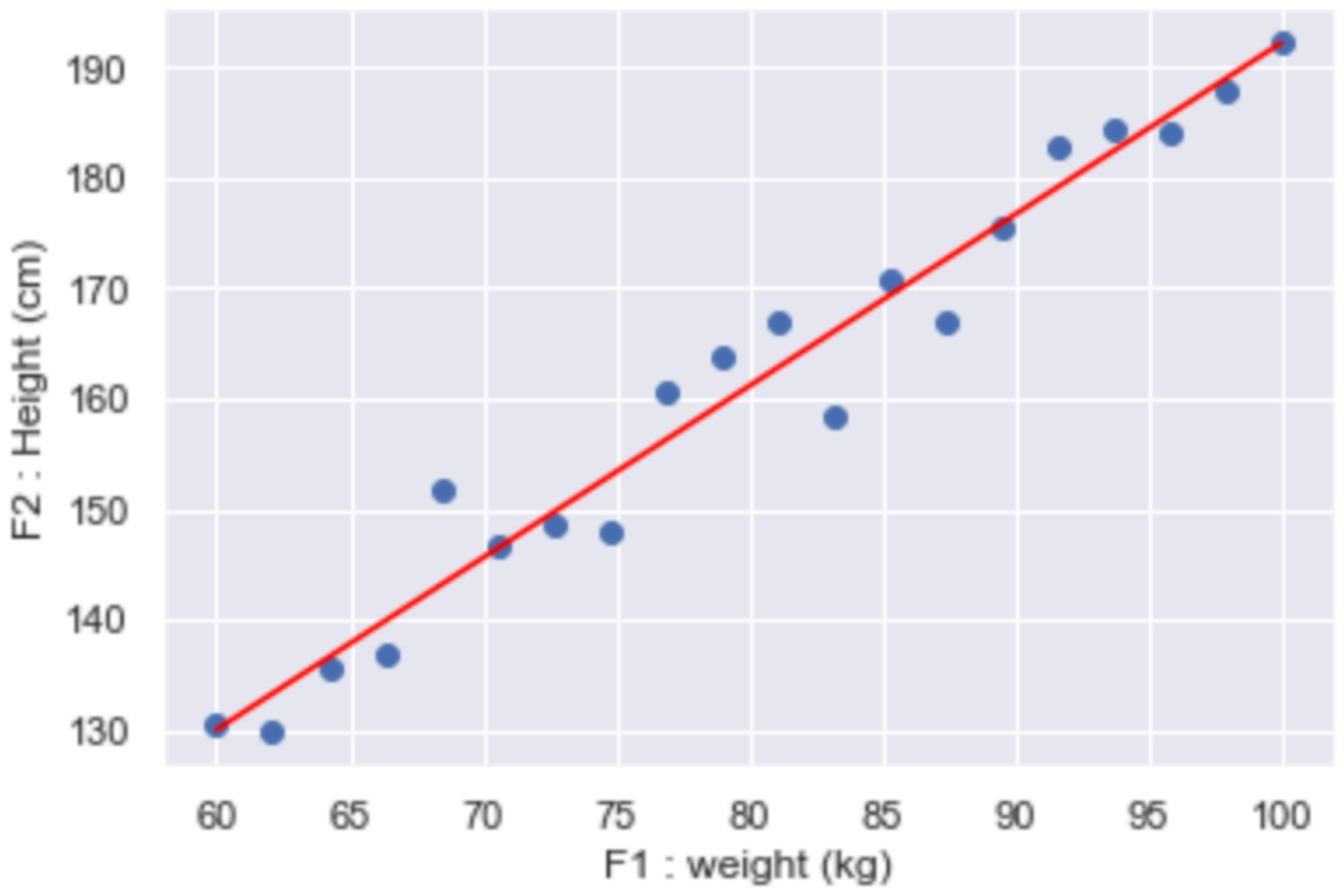
« Visualizing data using t-SNE », JMLR, Laurens et. al, 2008

3. Probabilistic dimensionality reduction

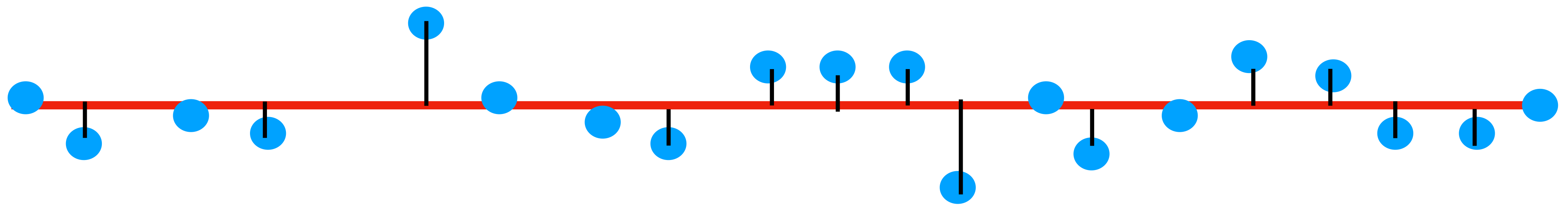
Dimensionality reduction : PCA

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data



The two features F1 and F2 have a positive correlation

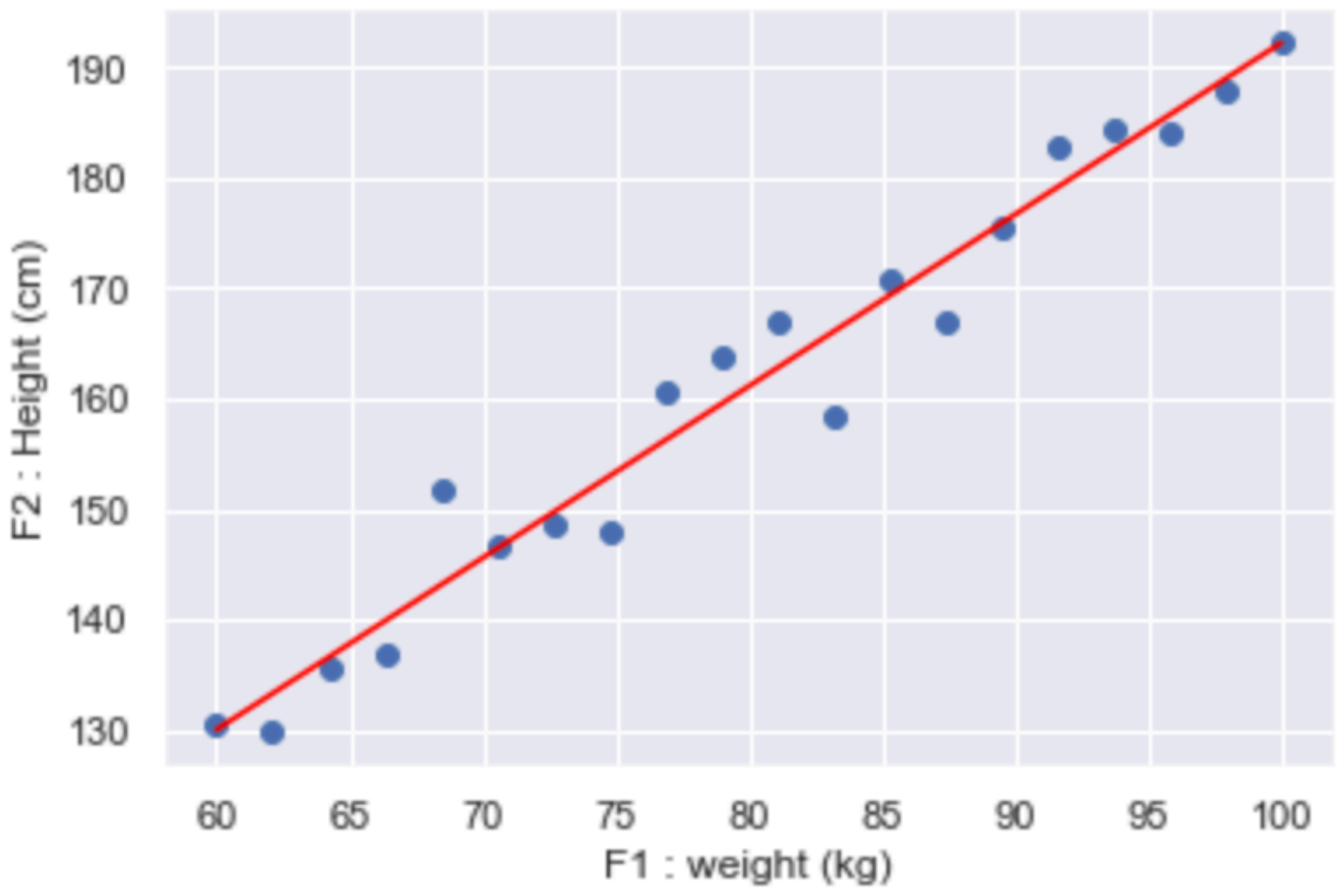


3. Probabilistic dimensionality reduction

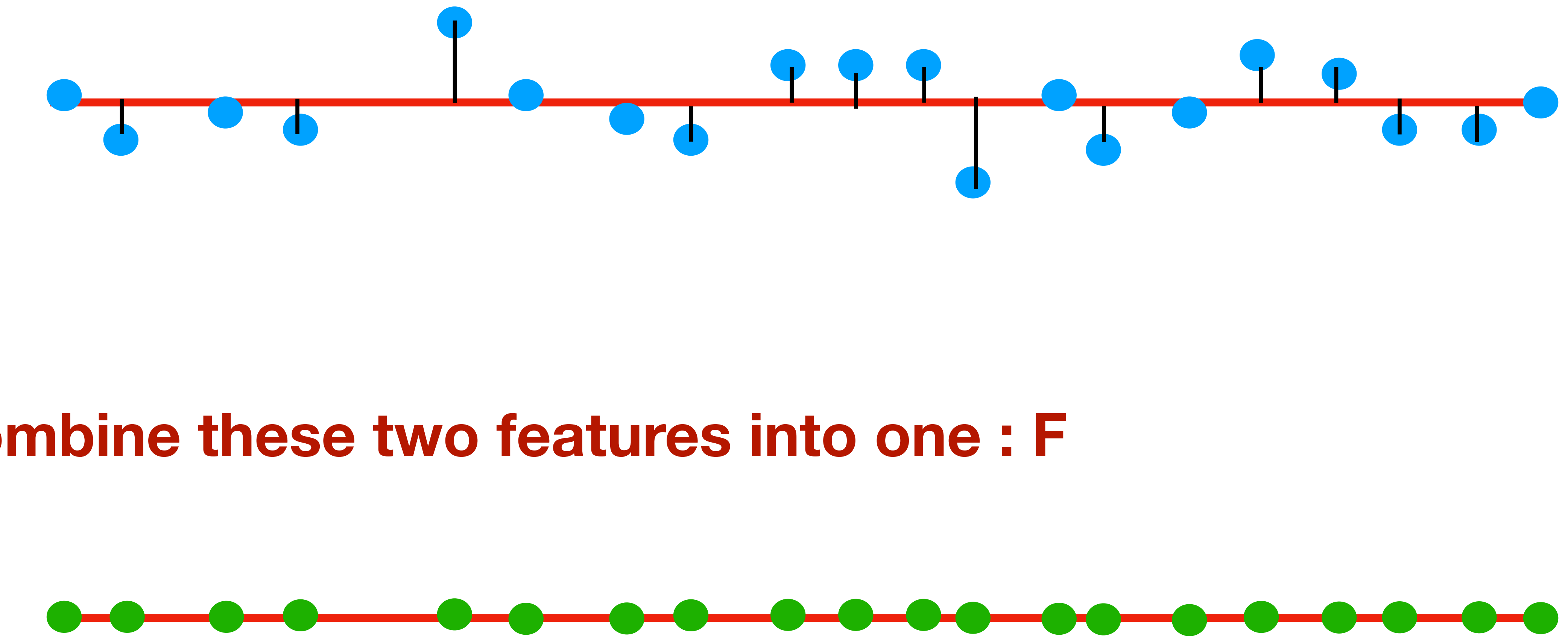
Dimensionality reduction : PCA

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data



The two features F1 and F2 have a positive correlation



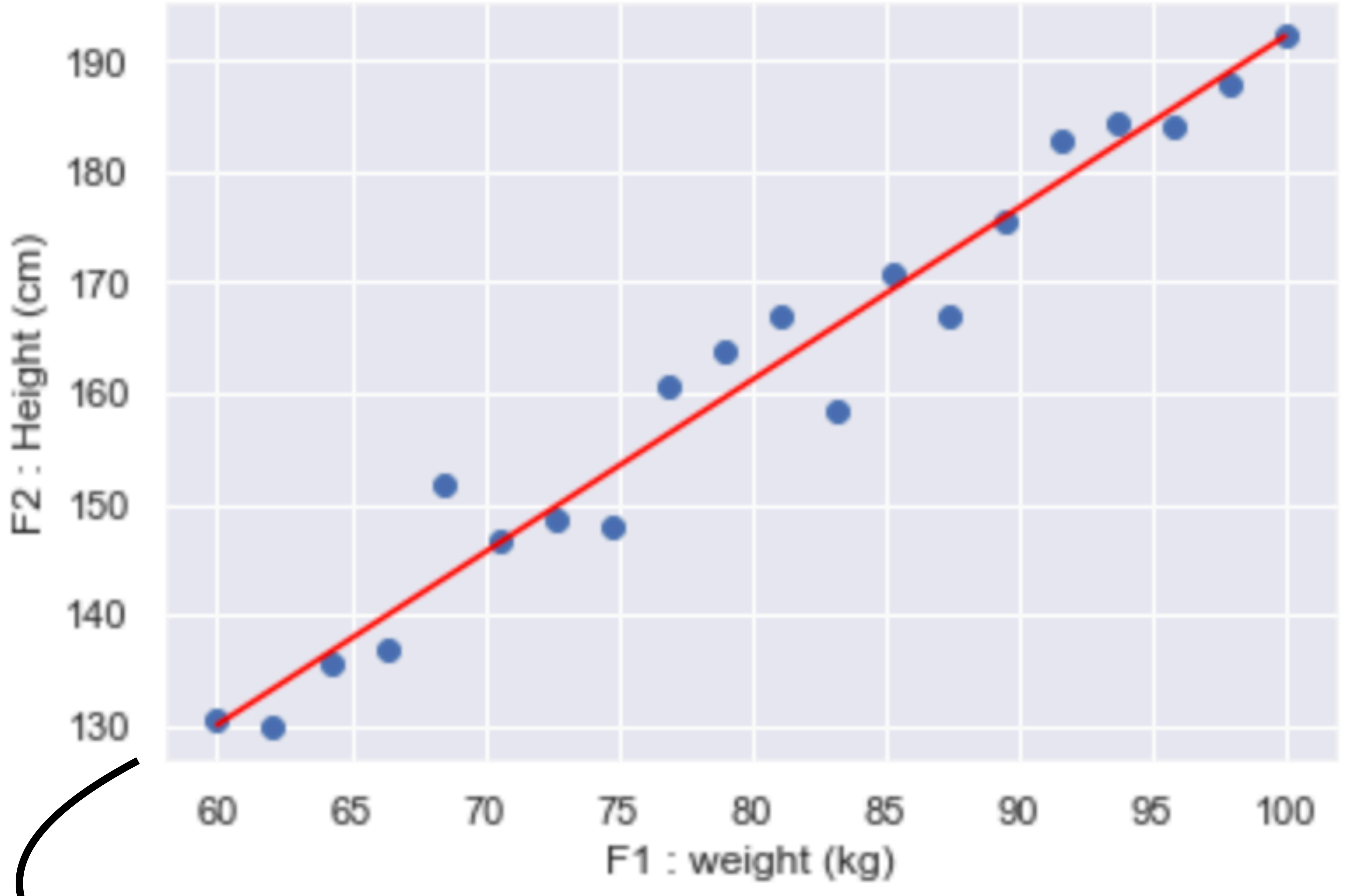
Combine these two features into one : F

3. Probabilistic dimensionality reduction

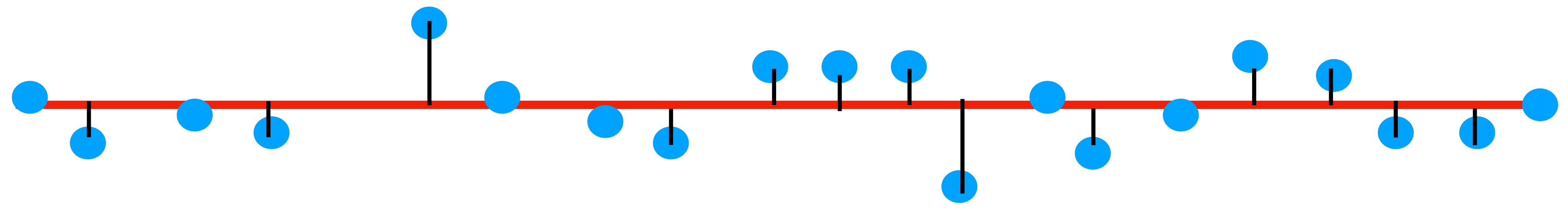
Dimensionality reduction : PCA

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

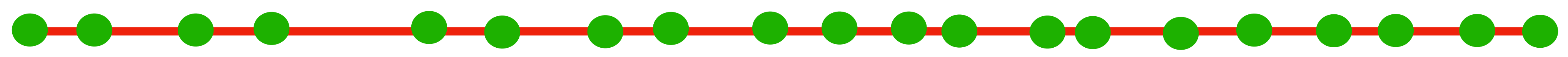
Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data



The two features F1 and F2 have a positive correlation



Combine these two features into one : F



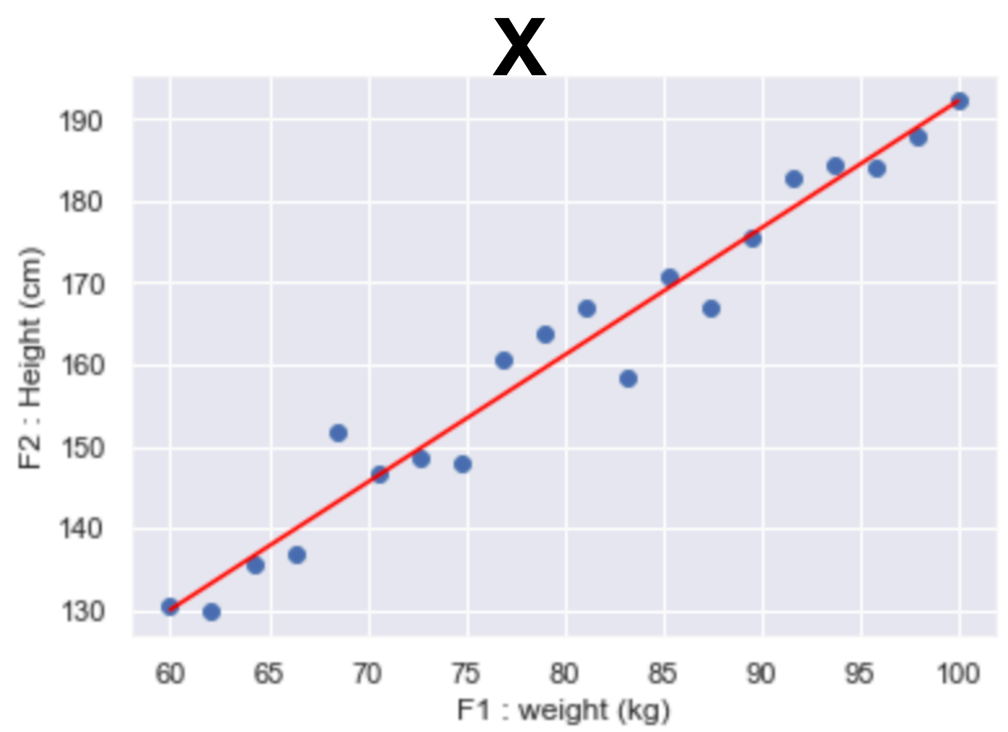
This line corresponds to the eigenvector associated to the greatest eigenvalue of the covariance matrix

3. Probabilistic dimensionality reduction

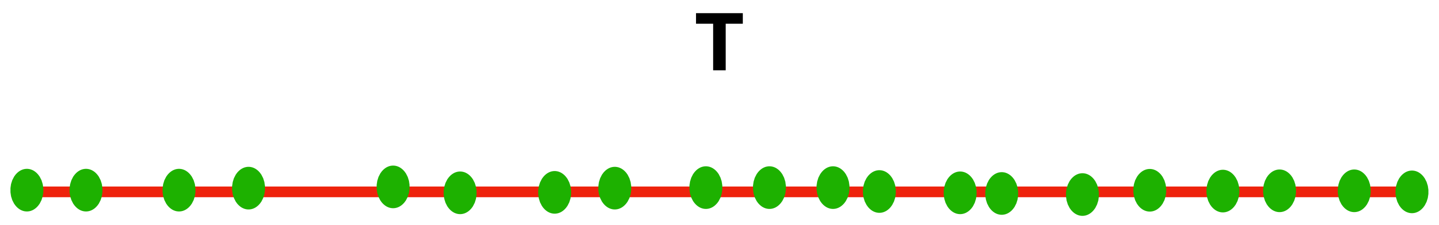
Dimensionality reduction : probabilistic PCA (PPCA)

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data



How do we **reduce** ?



Probabilistic PCA : a probabilistic point of view of PCA

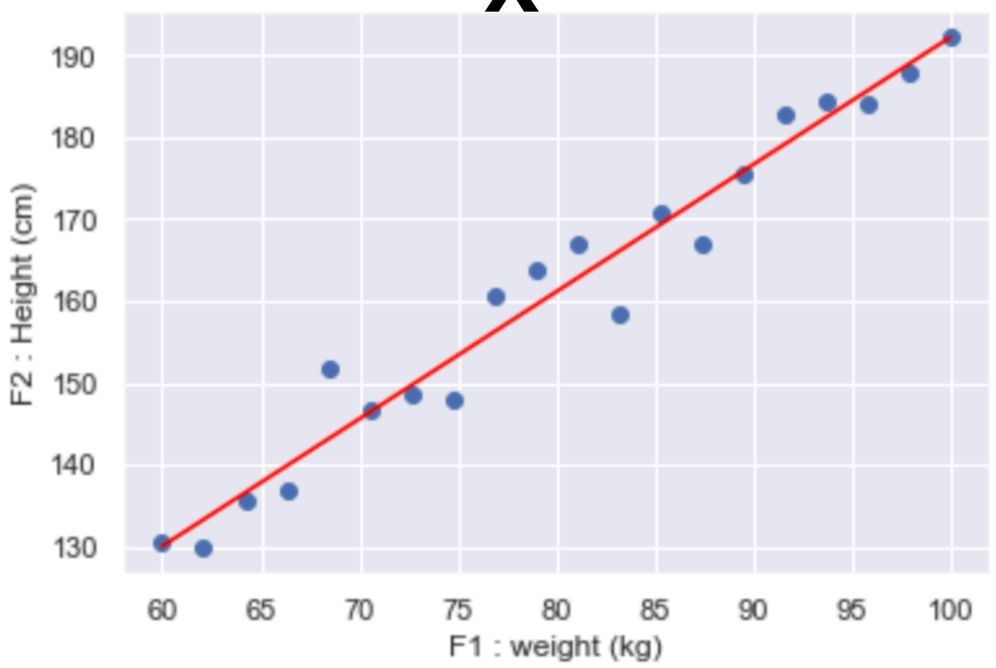
3. Probabilistic dimensionality reduction

Dimensionality reduction : probabilistic PCA (PPCA)

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

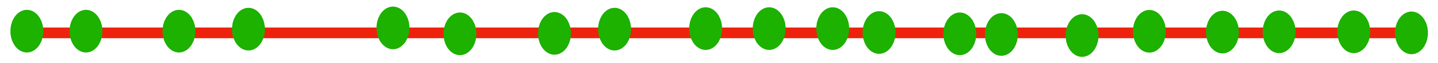
Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data

X



How do we **reduce** ?

T



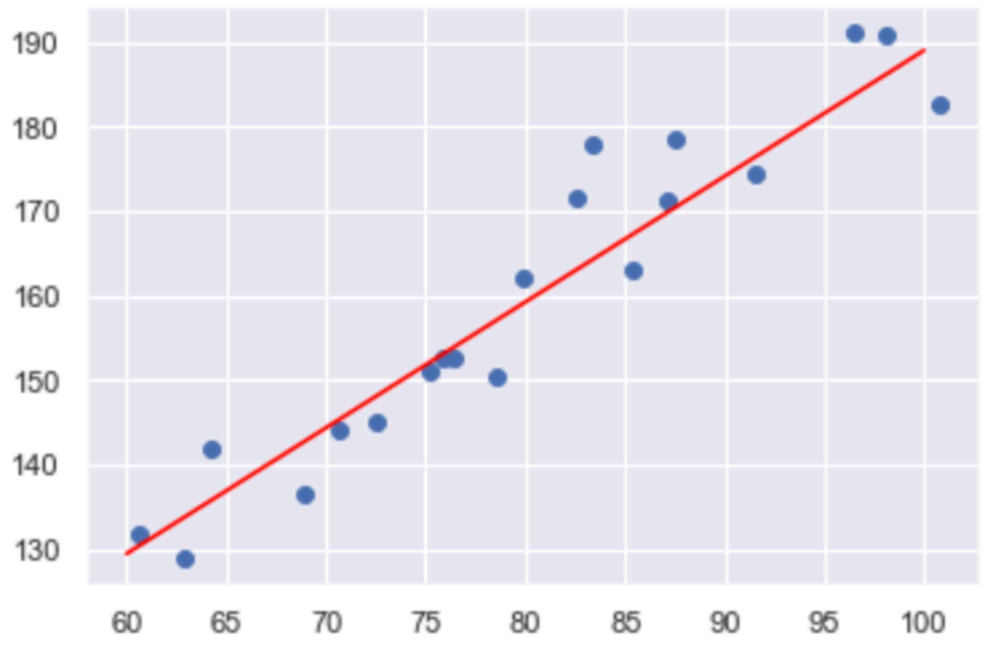
Probabilistic PCA : a probabilistic point of view of PCA

$$p(t_i) = \mathcal{N}(t_i \,|\, 0,\, I_2)$$

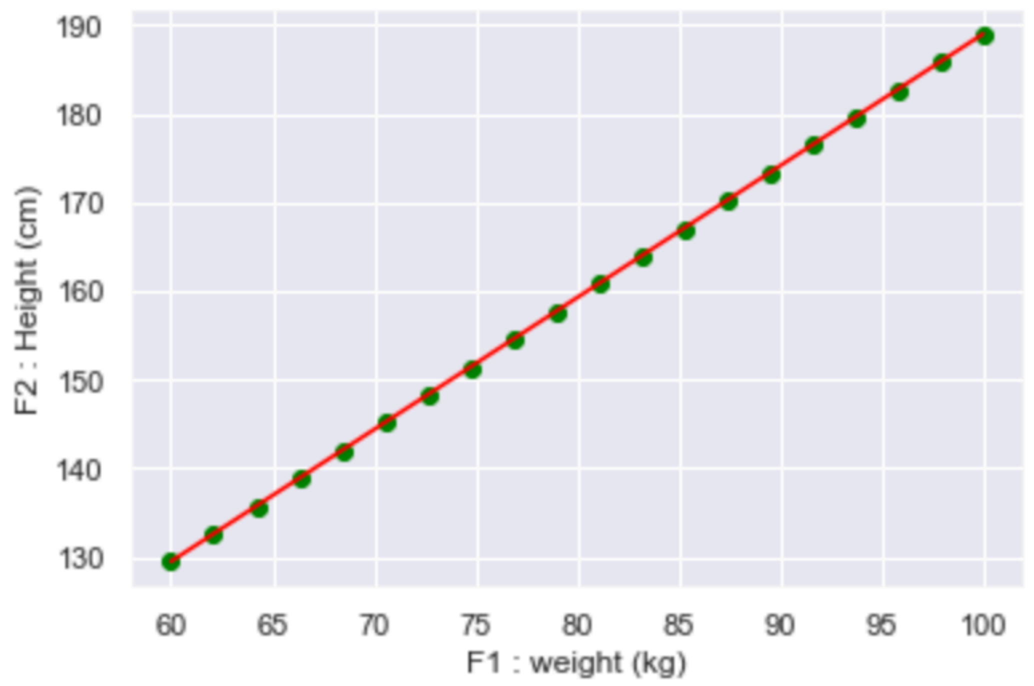
$$x_i = W\, t_i + b$$

$$x_i = W\, t_i + b + \epsilon_i \text{ with } \epsilon_i \sim \mathcal{N}(0, \Sigma)$$

$$p(x_i \,|\, t_i, \theta) = \ldots$$



How do we **generate** ?



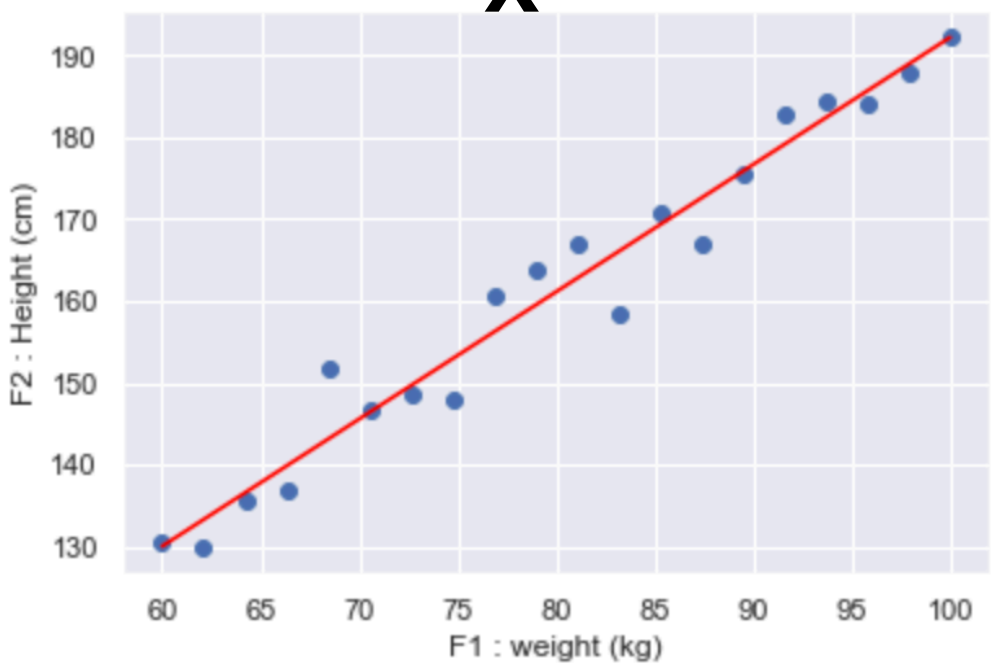
3. Probabilistic dimensionality reduction

Dimensionality reduction : probabilistic PCA (PPCA)

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

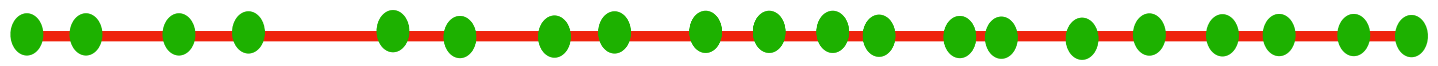
Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data

X



How do we **reduce** ?

T



Probabilistic PCA : a probabilistic point of view of PCA

$$p(t_i) = \mathcal{N}(t_i \mid 0,\, I_2)$$

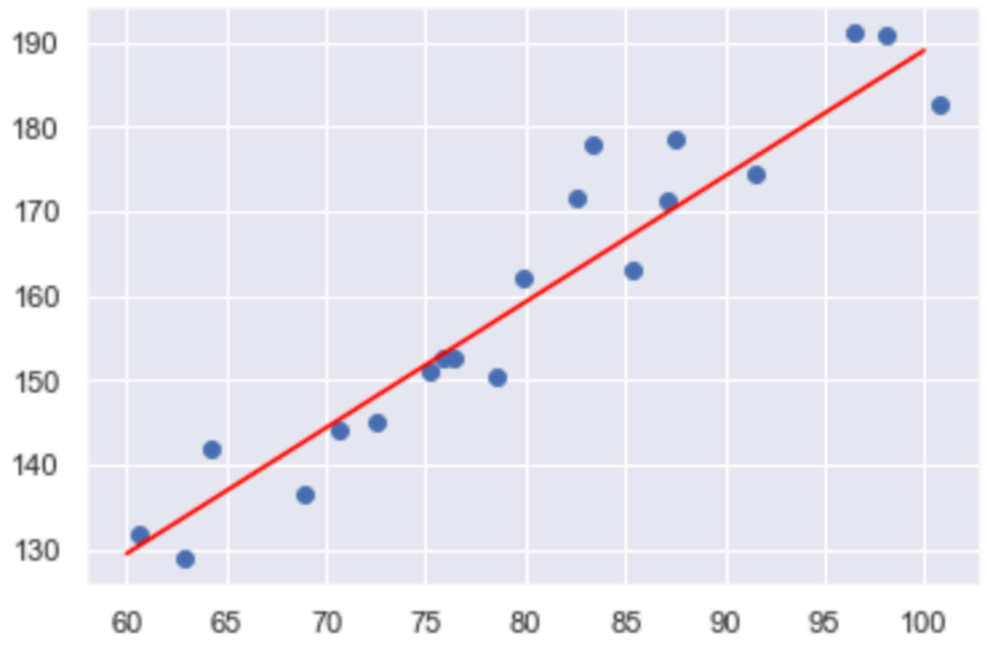
$$x_i = W\, t_i + b$$

$x_i = W\, t_i + b + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \Sigma)$

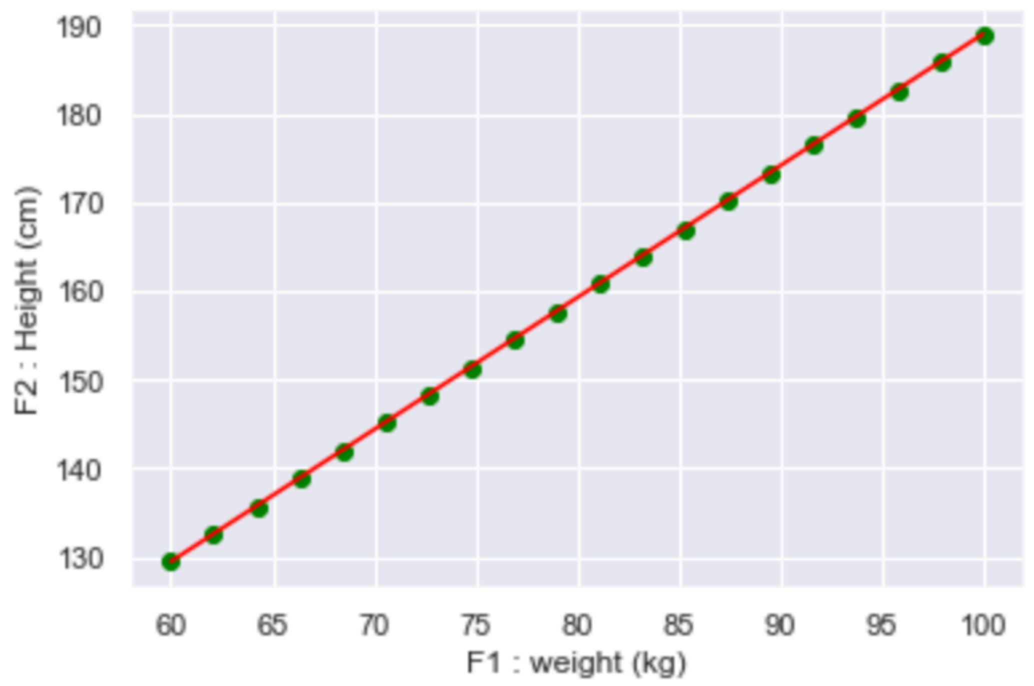
$$p(x_i \mid t_i, \theta) = \mathcal{N}(W t_i + b, \Sigma)$$

$$p(\mathrm{x} \mid \theta) = \prod_{i=1,\ldots,n} p(x_i \mid \theta)$$

$$= \prod_{i=1,\ldots,n} \int p(x_i \mid t_i, \theta) p(t_i) dt_i$$



How do we **generate** ?

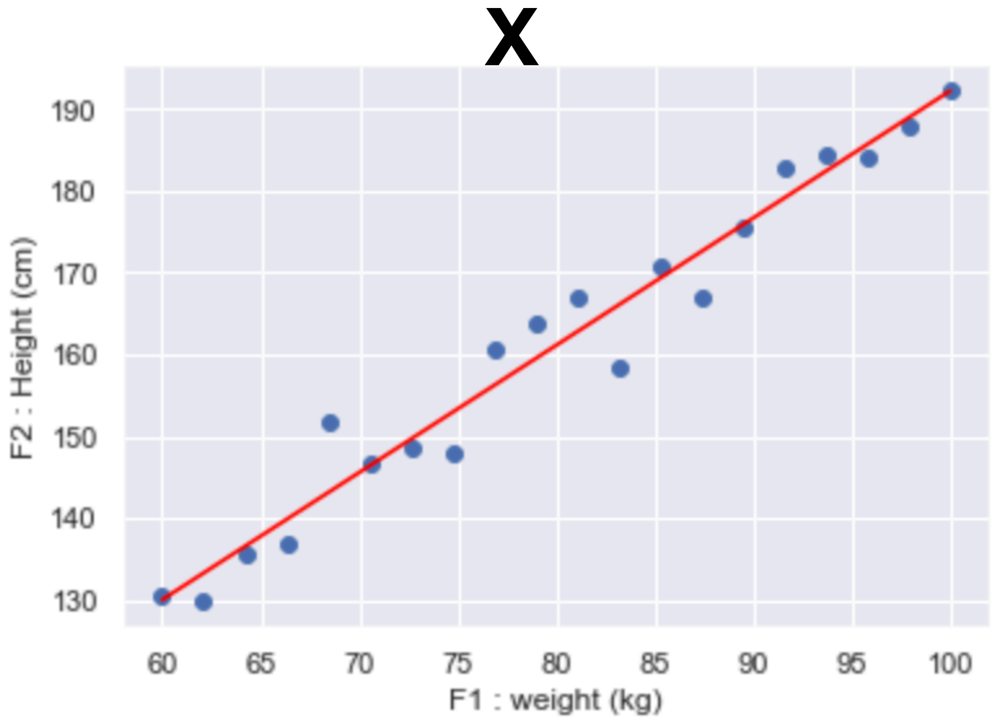


3. Probabilistic dimensionality reduction

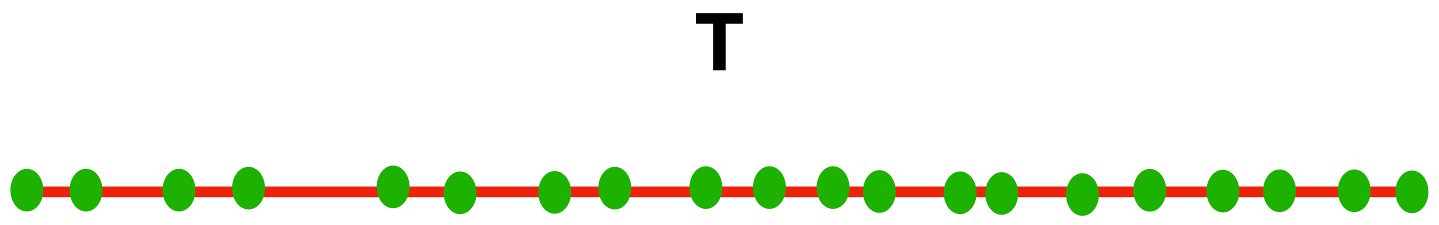
Dimensionality reduction : probabilistic PCA (PPCA)

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data



How do we **reduce** ?



Probabilistic PCA : a probabilistic point of view of PCA

$$p(t_i) = \mathcal{N}(t_i \,|\, 0,\, I_2)$$

$$x_i = W\, t_i + b$$

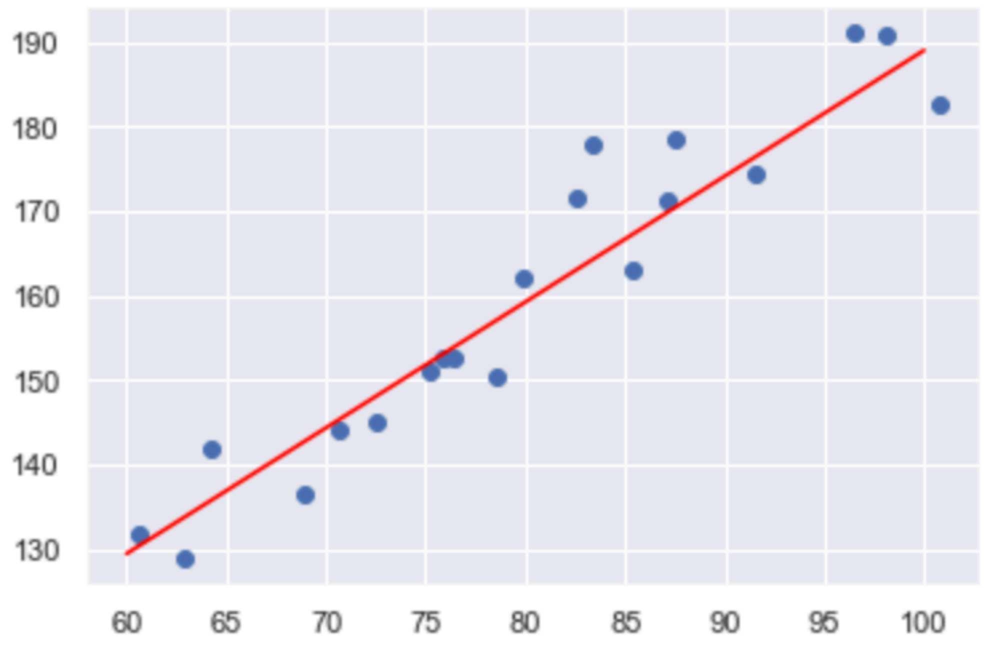
$x_i = W\, t_i + b + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \Sigma)$

$$p(x_i \,|\, t_i, \theta) = \mathcal{N}(Wt_i + b, \Sigma)$$

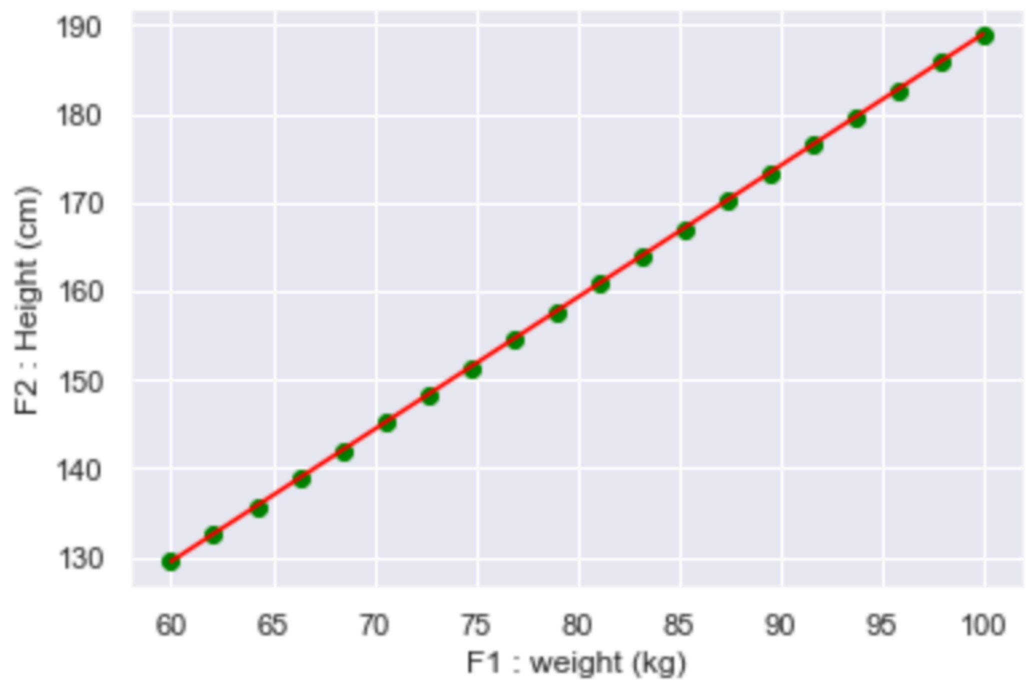
$$p(\mathrm{x} \,|\, \theta) = \prod_{i=1,\ldots,n} p(x_i \,|\, \theta)$$

$$= \prod_{i=1,\ldots,n} \int p(x_i \,|\, t_i, \theta) p(t_i) dt_i$$

Normal conjugacy !



How do we **generate** ?

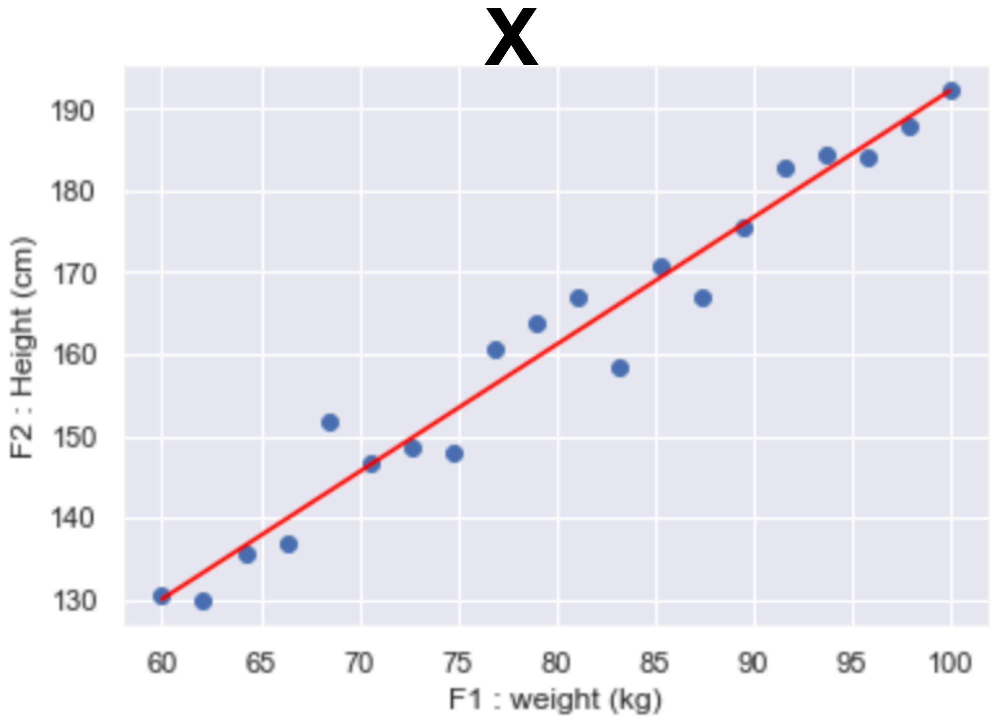


3. Probabilistic dimensionality reduction

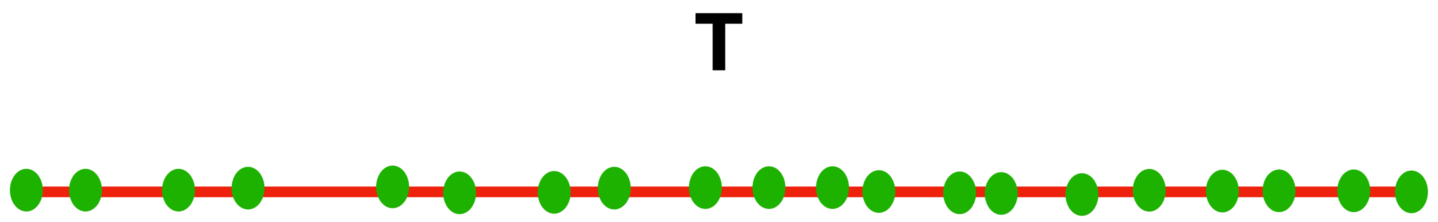
Dimensionality reduction : probabilistic PCA (PPCA)

Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

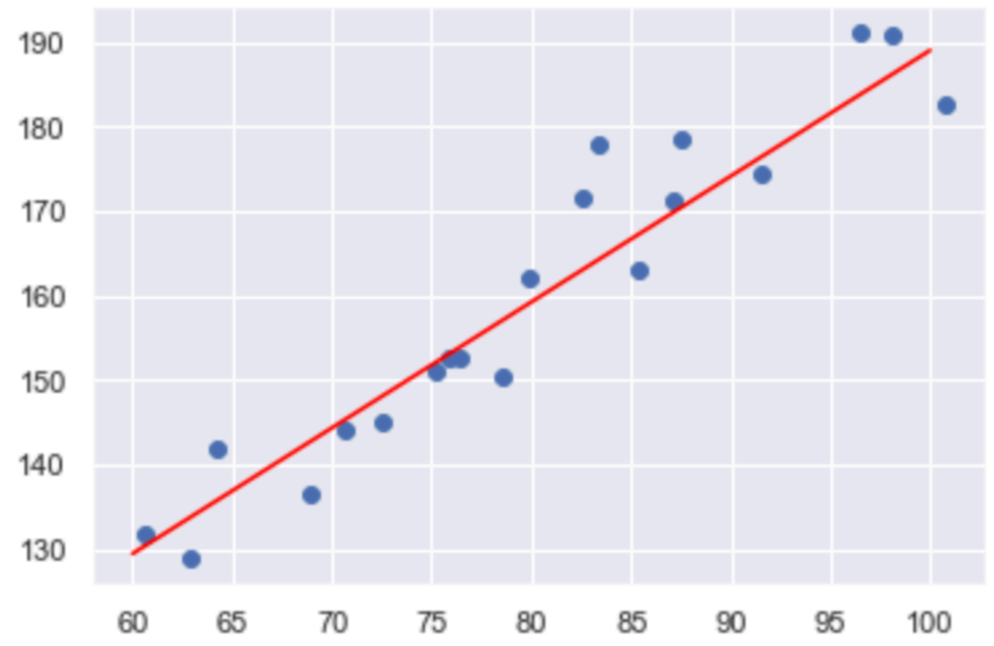
Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data



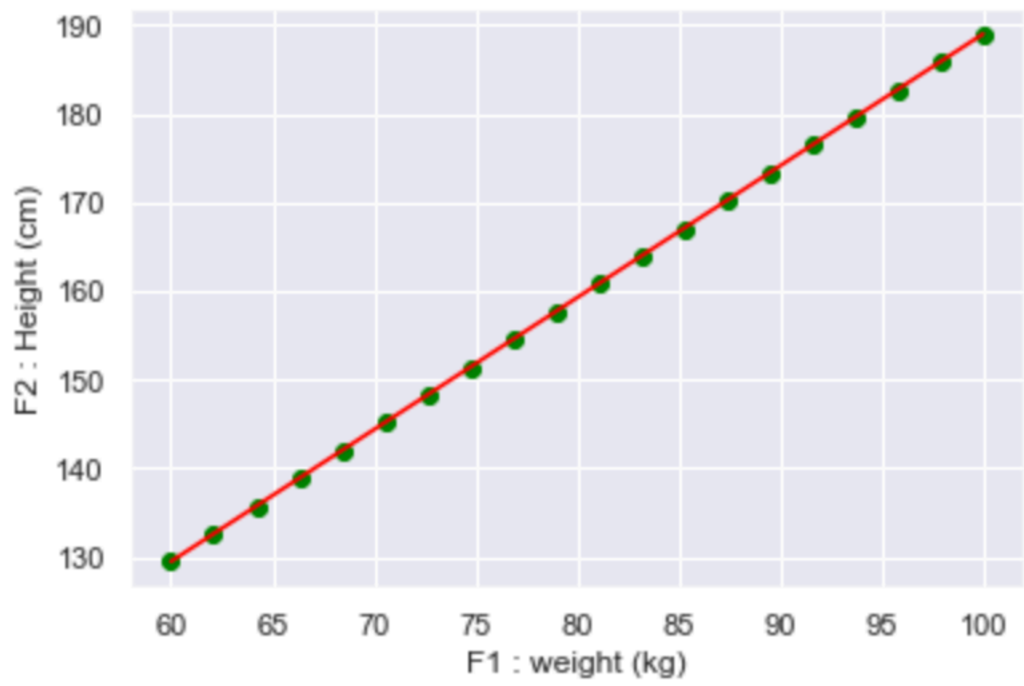
How do we **reduce** ?



Probabilistic PCA : a probabilistic point of view of PCA



How do we **generate** ?



$$p(t_i) = \mathcal{N}(t_i \mid 0, I_2)$$

$$x_i = W\, t_i + b$$

$$x_i = W\, t_i + b + \epsilon_i \text{ with } \epsilon_i \sim \mathcal{N}(0,\Sigma)$$

$$p(x_i \mid t_i, \theta) = \mathcal{N}(Wt_i + b, \Sigma)$$

$$p(X \mid \theta) = \prod_{i=1,\ldots,n} p(x_i \mid \theta)$$

$$= \prod_{i=1,\ldots,n} \int p(x_i \mid t_i, \theta) p(t_i) dt_i$$

Easy to do EM here !

Normal conjugacy !

3. Probabilistic dimensionality reduction

Dimensionality reduction : probabilistic PCA (PPCA)

Probabilistic PCA : a probabilistic point of view of PCA

EM for PPCA :

E-step : $q(t_i) = p(t_i \mid x_i, \theta) = \dfrac{p(x_i \mid t_i, \theta)\, p(t_i)}{\text{constant}}$ prior conjugacy

M-step : $\max\limits_{\theta} \leftarrow \mathbb{E}_{q(T)} \sum_i \log p(x_i \mid t_i, \theta)\, p(t_i)$

$= \sum_i \mathbb{E}_{q(t_i)} \log \left(\frac{1}{\text{const}} e^{\cdots} e^{\cdots} \right)$

$= \sum_i \log \left(\frac{1}{\text{const}} \right) + \underbrace{\sum_i \mathbb{E}_{q(t_i)} \log \left(e^{-\frac{(x - Wt_i - b)^2}{2\sigma^2}} e^{-\frac{t_i^2}{2}} \right)}_{\text{quadratic function on } t,\ \text{so we can do it analytically}}$

Some cool things with PPCA :

- We can fill **missing values**
- **Hyperparameters** tuning
- We can do **mixture of PPCA**

« Probabilistic Principal Component Analysis », JMLR, Michael E. Tipping et. al, 1999

4 Applications and examples : notebook

Application and examples

website : https://curiousml.github.io/

// EPITA - École pour l'informatique et les techniques avancées (2020 - ...)

- Master of Science in Artificial Intelligence Systems : **Bayesian Machine Learning** by François HU
  - **Training session / prerequisite :** [Statistics with python], [Data]
  - **Lecture 1** : [Bayesian statistics]
  - **Practical work 1** : [Conjugate distributions] [Correction]
  - **Lecture 2** : [Latent Variable models and EM-algorithm]
  - **Practical work 2** : [probabilistic K-means and probabilistic PCA] [Correction]
  - **Lecture 3** : (soon available)
  - **Practical work 3** : (soon available)
  - **Lecture 4** : (soon available)
  - **Practical work 4** : (soon available)
  - **Lecture 5** : (soon available)

TODO

! Road map

Bayesian statistics (03/05/21)

1

Bayesian perspective :

Likelihood

Prior distribution

$$P(\theta | X) = \frac{P(X, \theta)}{P(X)} = \frac{P(X|\theta) \cdot P(\theta)}{P(X)}$$

Posterior distribution

Evidence

Hard to compute !

θ parameters

X observations

Exemple :
Naive Bayes classifier,
Linear regression, ….

MAP : $\arg\max_{\theta} P(X|\theta) \cdot P(\theta)$

Conjugate distribution

Pros :
- exact posterior

Cons :
- conjugate prior maybe inadequate

Latent variable models (17/05/21)

2

Hidden variable models :

$$P(X|\theta) = \sum_{t \in T_{indexes}} P(X, T = t|\theta)$$

$$P(X, T|\theta) = P(X|T, \theta)P(T|\theta)$$

Exemple :
GMM, K-means, PCA/PPCA

Pros :
- fewer parameters / simpler models
- hidden variable sometimes meaningful
- clustering / dimensionality reduction

Cons :
- harder to work with
- requires math
- only local maximum or saddle point
- EM : the posterior of T could be intractable

Variational Inference (31/05/21)

3

Markov Chain Monte Carlo (07/06/21)

4

Extensions (14/06/21)

5