Bayesian Machine Learning

May 2024 - François HU
https://curiousml.github.io/

Outline

1 **Bayesian statistics**

2 **Latent variable models**
- Latent variable models and EM algorithm
- Probabilistic clustering
- Probabilistic dimensionality reduction

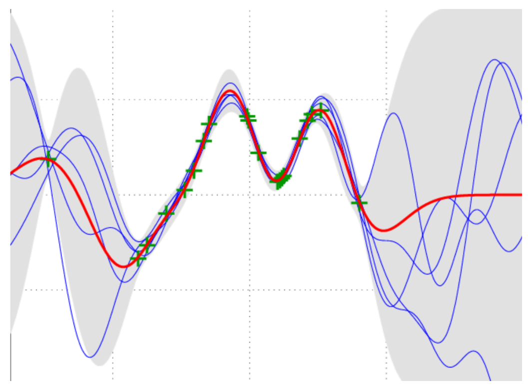
3 **Variational Inference**

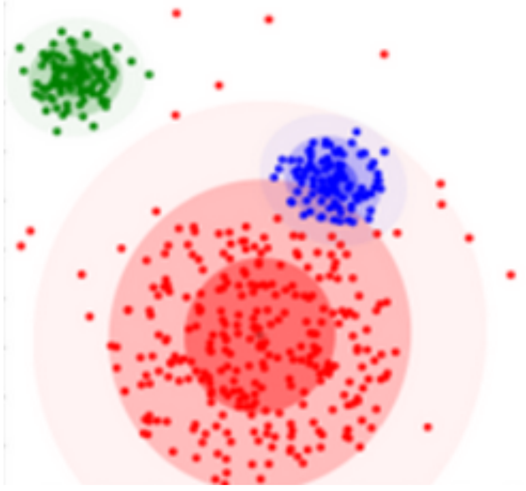
4 **Markov Chain Monte Carlo**

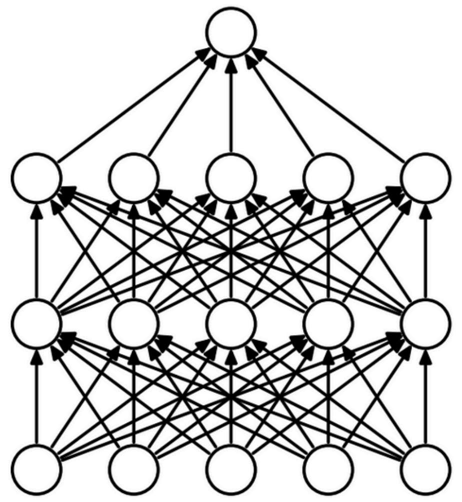
5 **Extensions and oral presentations**

0 Evaluation & conjugate prior

Evaluation (1/2)
Group project







- The evaluation will consist of a **group project (4 students max)** based on a research article

- For the last lecture, each student will send me the **codes (only in Python!** Some of the paper's implementations are in R or Matlab, but you need to adapt them**)** and give an **oral presentation** in front of the class. Even if the article is mostly theoretical, each presentation should be understandable by other students (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 (highly recommended, it should be cited at the end of your presentation) to boost your knowledge.

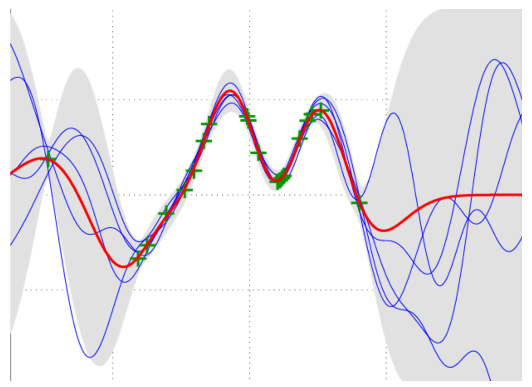
- The **evaluation** is as follows :

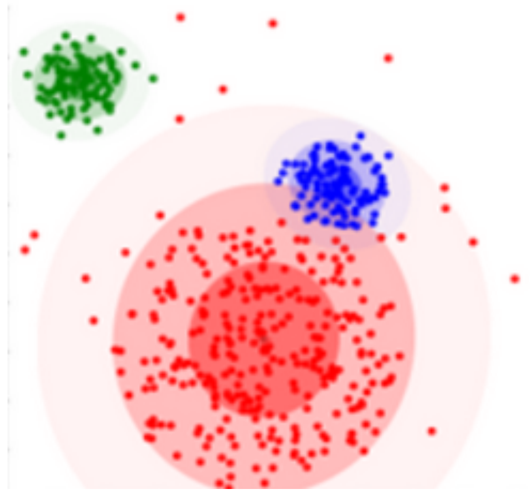
  - **40% on the clarity of the code** (example : many comments, along with understandable variables/functions names. You can use Jupyter Notebook which might have the advantage to be easy to read for the users). When I run your code, it should be easy to run and easy to understand :)

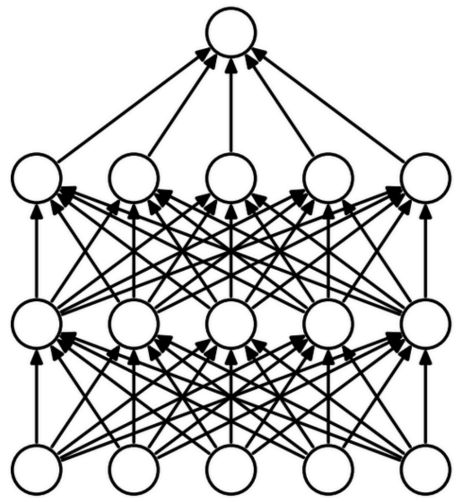
  - **60% on the clarity of the oral presentation**. Less maths but more experimentations and intuitions. At the beginning a big introduction is expected in order to be understandable by other groups.

Evaluation (2/2)

Group project







1. **Project Interpretability**:
   main paper « **DAG-GNN**: DAG structure learning with graph neural networks »

2. **Project Fairness**:
   main paper: « Fair Data Adaptation with Quantile Preservation »
   R package: « **fairadapt**: Causal Reasoning for Fair Data Pre-processing »

3. **Project Uncertainty**:
   main paper: « Dropout as a bayesian approximation: Representing model uncertainty in deep learning »

4. **Project Topic modeling**:
   main paper: « Mixing dirichlet topic models and word embeddings to make **lda2vec** »
   open question: Propose a way to automatically generate a topic's title. Implement it.

5. **Project Missing values:**
   main paper: « What's a good imputation to predict with missing values? »

Conjugate priors: Exercices

Gamma case

Exercice (left as an exercice, correction in the next lecture)

$$P(\gamma|x) = \frac{\mathcal{N}(x|\mu,\gamma^{-1}) \times P(\gamma)}{P(x)}$$

$P(\gamma)$: $\Gamma(\gamma|\alpha_{prior},\beta_{prior})$

$P(\gamma|x)$: $\Gamma(\gamma|\alpha_{posterior},\beta_{posterior})$

$\Gamma(\gamma|\alpha_{prior}+1/2,\ \beta_{prior}+(x-\mu)^2/2)$

Gamma distribution

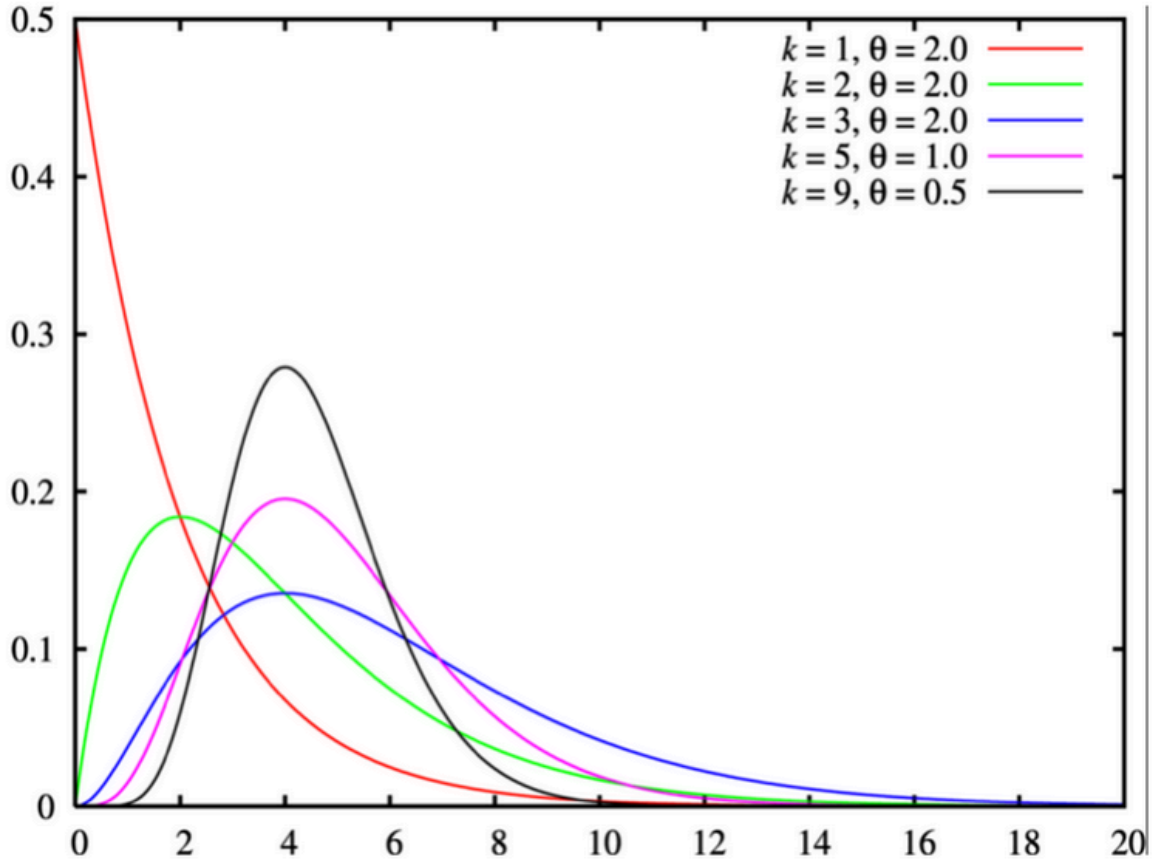
PDF : $\Gamma(x|\alpha,\beta) = \dfrac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$ with $x,\alpha,\beta > 0$

$\Gamma(\alpha) = (\alpha-1)!$

mean : $\mathbb{E}[x] = \dfrac{\alpha}{\beta}$

variance : $V(x) = \dfrac{\alpha}{\beta^2}$

mode : $Mode\,[x] = \dfrac{\alpha-1}{\beta}$



What we want to compute : $p(\text{parameters} \mid \text{data}) \propto p(\text{data} \mid \text{parameters}) \times p(\text{parameters})$

- $p(\text{data} \mid \text{parameters}) = \mathcal{N}(\overset{\text{deterministic}}{x} \mid \overset{\text{deterministic}}{\mu}, \overset{\text{random}}{\gamma^{-1}}) = \dfrac{\sqrt{\gamma}}{\sqrt{2\pi}} e^{-\gamma \frac{(x-\mu)^2}{2}} \propto \sqrt{\gamma}\, e^{-\gamma \frac{(x-\mu)^2}{2}}$

- $p(\text{parameters}) = \Gamma(\overset{\text{random}}{\gamma} \mid \overset{\text{deterministic}}{\alpha_{prior}}, \overset{\text{deterministic}}{\beta_{prior}}) = \dfrac{\beta_{prior}^{\alpha_{prior}}}{\Gamma(\alpha_{prior})} \times \gamma^{\alpha_{prior}-1} e^{-\gamma\beta_{prior}} \propto \gamma^{\alpha_{prior}-1} e^{-\gamma\beta_{prior}}$

So: $p(\text{parameters} \mid \text{data}) \propto \gamma^{1/2} e^{-\gamma\frac{(x-\mu)^2}{2}} \times \gamma^{\alpha_{prior}-1} e^{-\gamma\beta_{prior}}$

$$\propto \gamma^{\frac{1}{2}+\alpha_{prior}-1}\, e^{-\gamma\left(\beta_{prior} + \frac{(x-\mu)^2}{2}\right)}$$

$$p(\text{parameters} \mid \text{data}) = \boxed{\Gamma\left(\gamma \mid \underbrace{\alpha_{prior}+\tfrac{1}{2}}_{\alpha_{posterior}},\ \underbrace{\beta_{prior} + \tfrac{(x-\mu)^2}{2}}_{\beta_{posterior}}\right)}$$

Conjugate priors: Exercices

Beta case

Exercice (left as an exercice, correction in the next lecture)

$B(\theta \mid \alpha_{posterior}, \beta_{posterior})$

$B(\theta \mid n_1 + \alpha_{prior}, n_0 + \beta_{prior})$

$$P(\theta \mid x) = \frac{Ber(x \mid \theta) \times P(\theta)}{P(x)}$$

$\theta^{n_1} \cdot (1-\theta)^{n_0}$

$B(\theta \mid \alpha_{prior}, \beta_{prior})$

Beta distribution

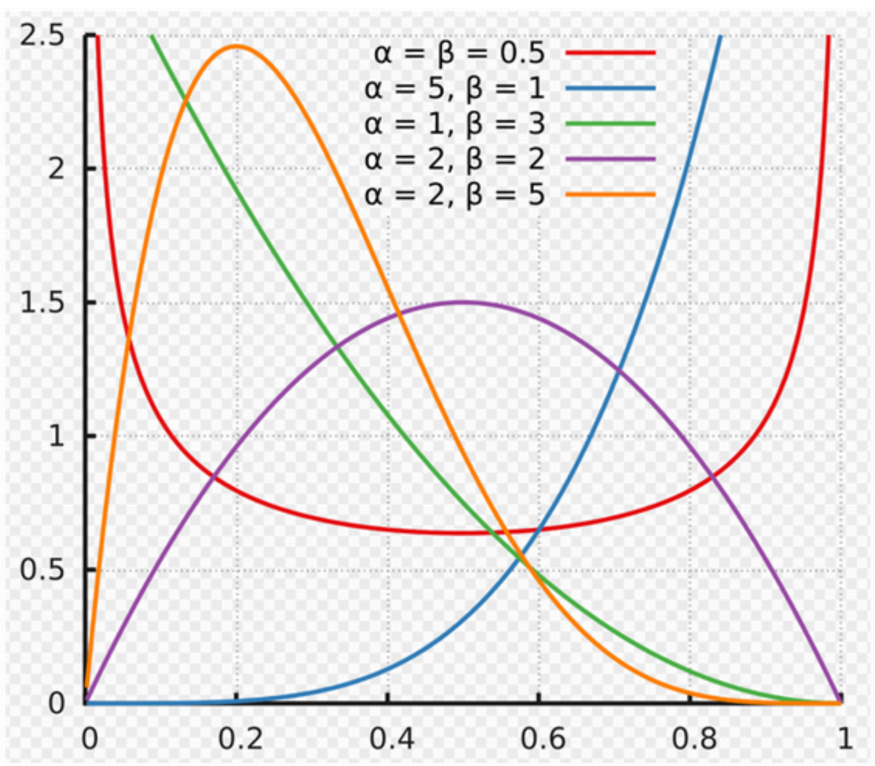
PDF : $B(x \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$ with $\alpha, \beta > 0$ and $x \in [0,1]$

$$B(\alpha, \beta) = \frac{\Gamma(\alpha) \cdot \Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

mean : $\mathbb{E}[x] = \frac{\alpha}{\alpha+\beta}$

variance : $V(x) = \frac{\alpha\beta}{(\alpha+\beta)^2 \cdot (\alpha+\beta-1)}$

mode : $Mode\,[x] = \frac{\alpha-1}{\alpha+\beta-2}$



What we want to compute : $p(\text{parameters} \mid \text{data}) \propto p(\text{data} \mid \text{parameters}) \times p(\text{parameters})$

- $p(\text{data} \mid \text{parameters}) = Ber(x \mid \theta) = \binom{n}{x} \theta^{x} (1-\theta)^{n-x} \propto \theta^{\cancel{x} \to n_1} (1-\theta)^{\cancel{n-x} \to n_0}$

- $p(\text{parameters}) = B(\theta \mid \alpha_{prior}, \beta_{prior}) = \dfrac{\theta^{\alpha_{prior}-1}(1-\theta)^{\beta_{prior}-1}}{B(\alpha_{prior}, \beta_{prior})} \propto \theta^{\alpha_{prior}-1}(1-\theta)^{\beta_{prior}-1}$

So: $p(\text{parameters} \mid \text{data}) \propto \theta^{n_1}(1-\theta)^{n_0} \times \theta^{\alpha_{prior}-1}(1-\theta)^{\beta_{prior}-1}$

$\propto \theta^{n_1+\alpha_{prior}-1}(1-\theta)^{n_0+\beta_{prior}-1}$

$$p(\text{parameters} \mid \text{data}) = \boxed{B(\theta \mid \underbrace{n_1+\alpha_{prior}-1}_{\alpha_{posterior}}, \underbrace{n_0+\beta_{prior}-1}_{\beta_{posterior}})}$$

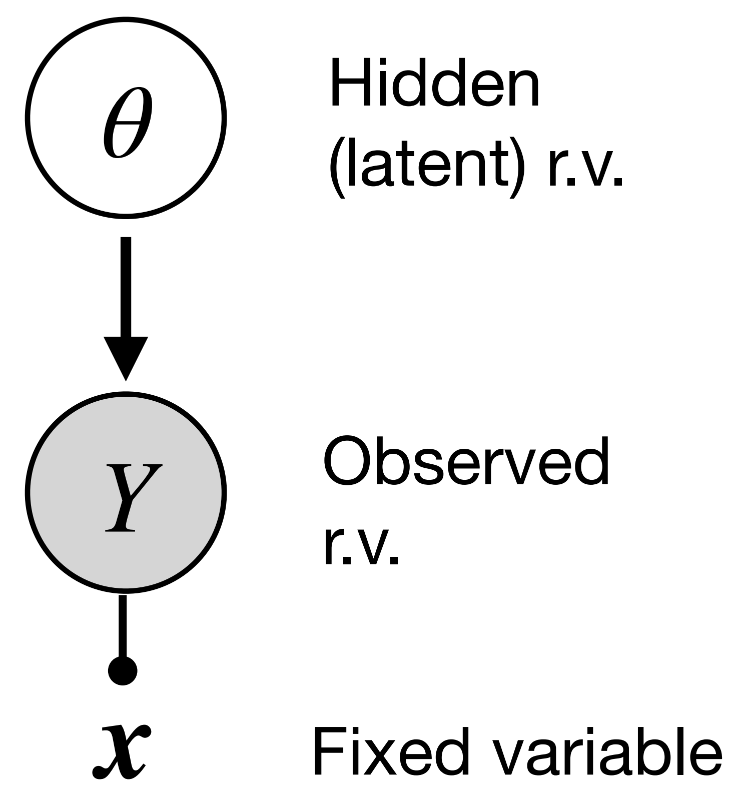
1 Latent variable models and mixture models

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

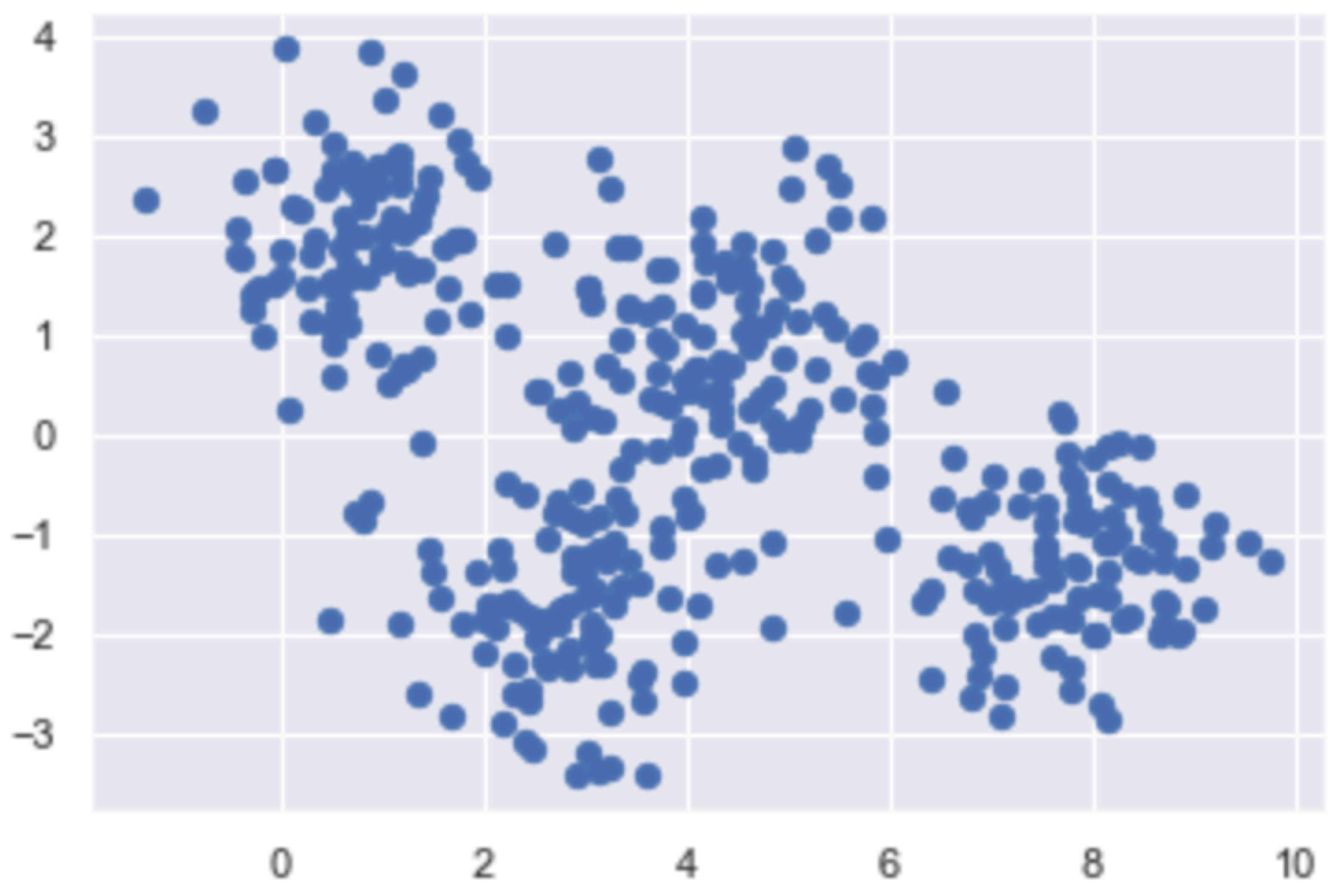
1. Latent Variable Models

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



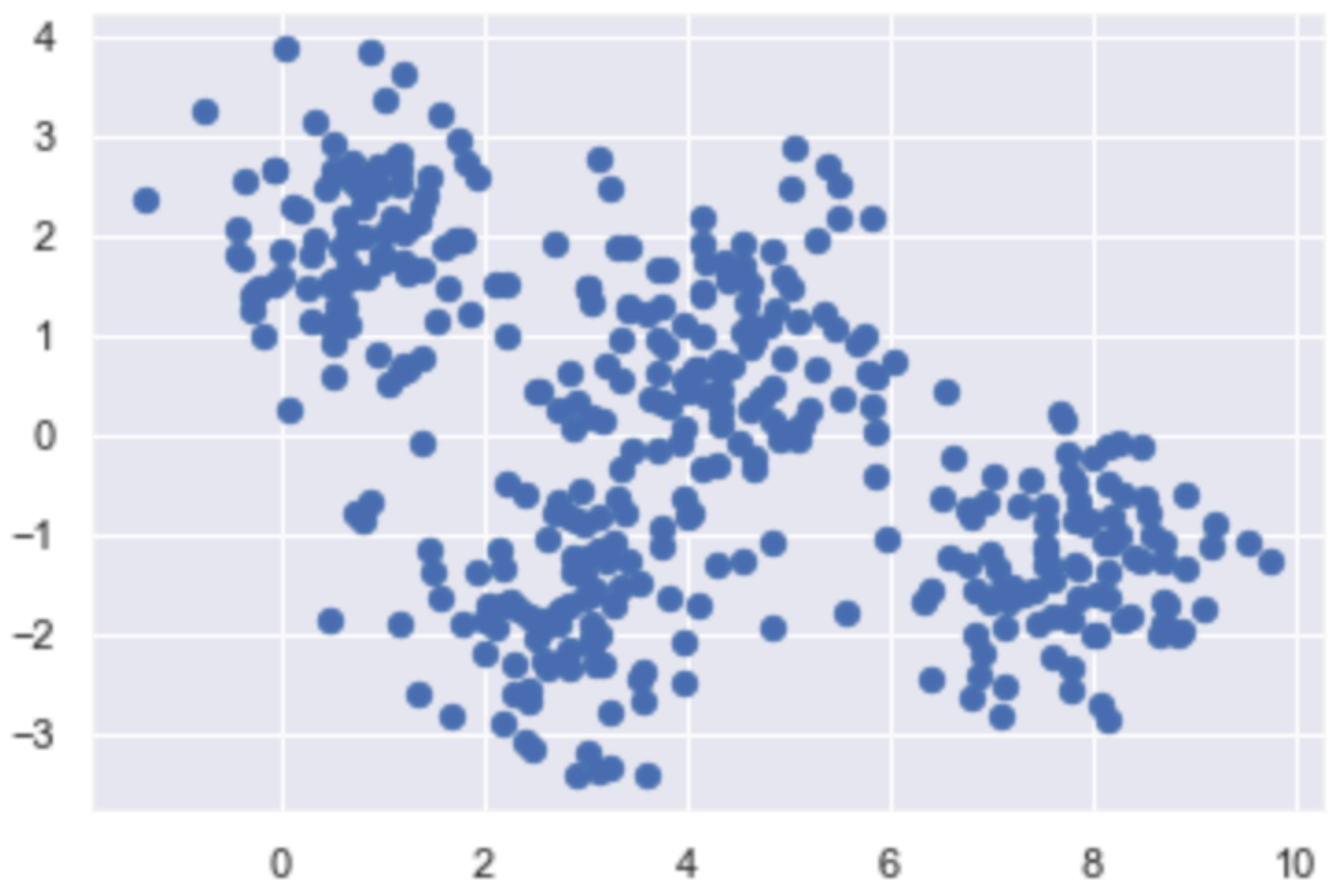
1. Latent Variable Models

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 !

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

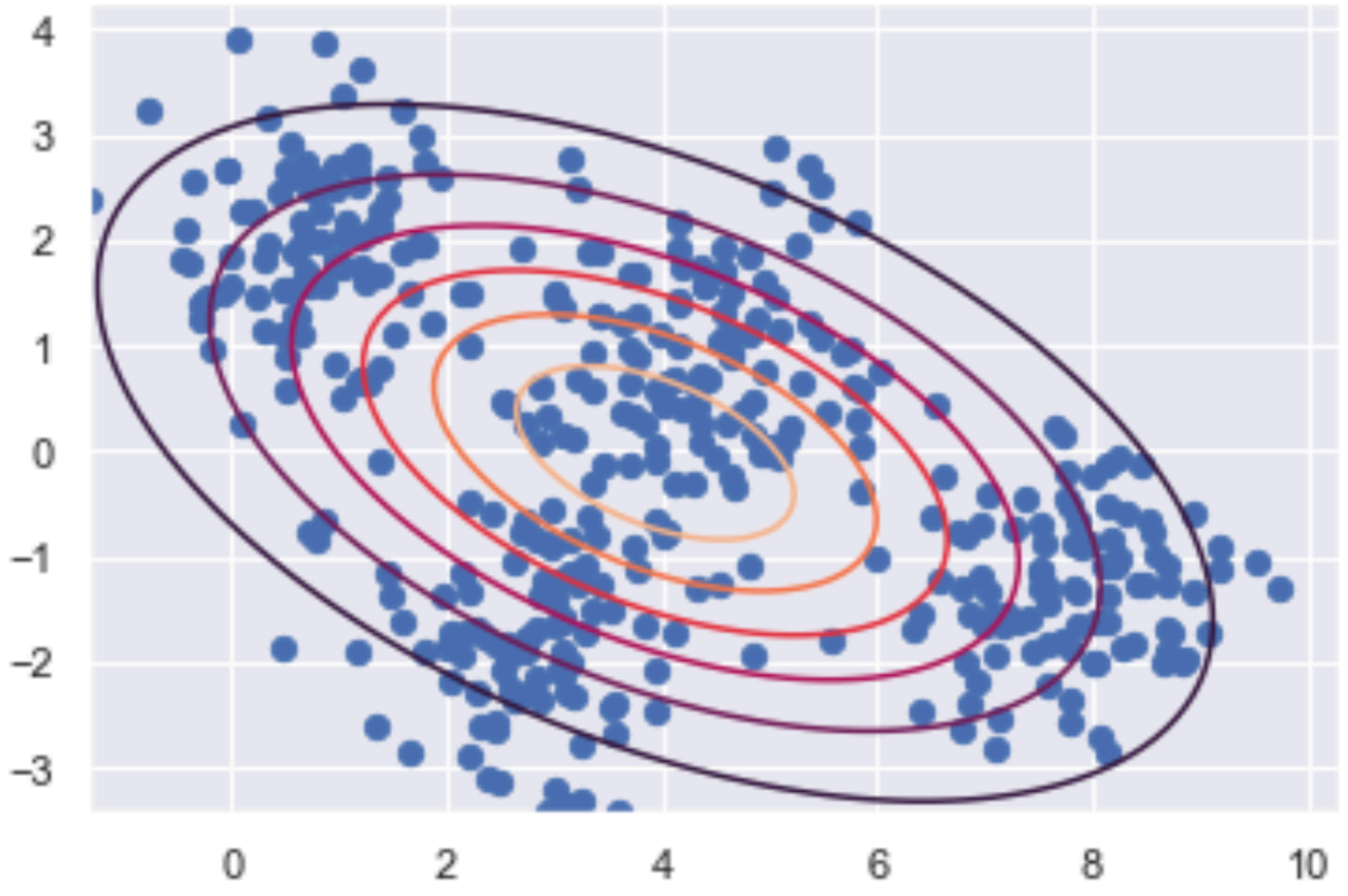
1. Latent Variable Models

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 !

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

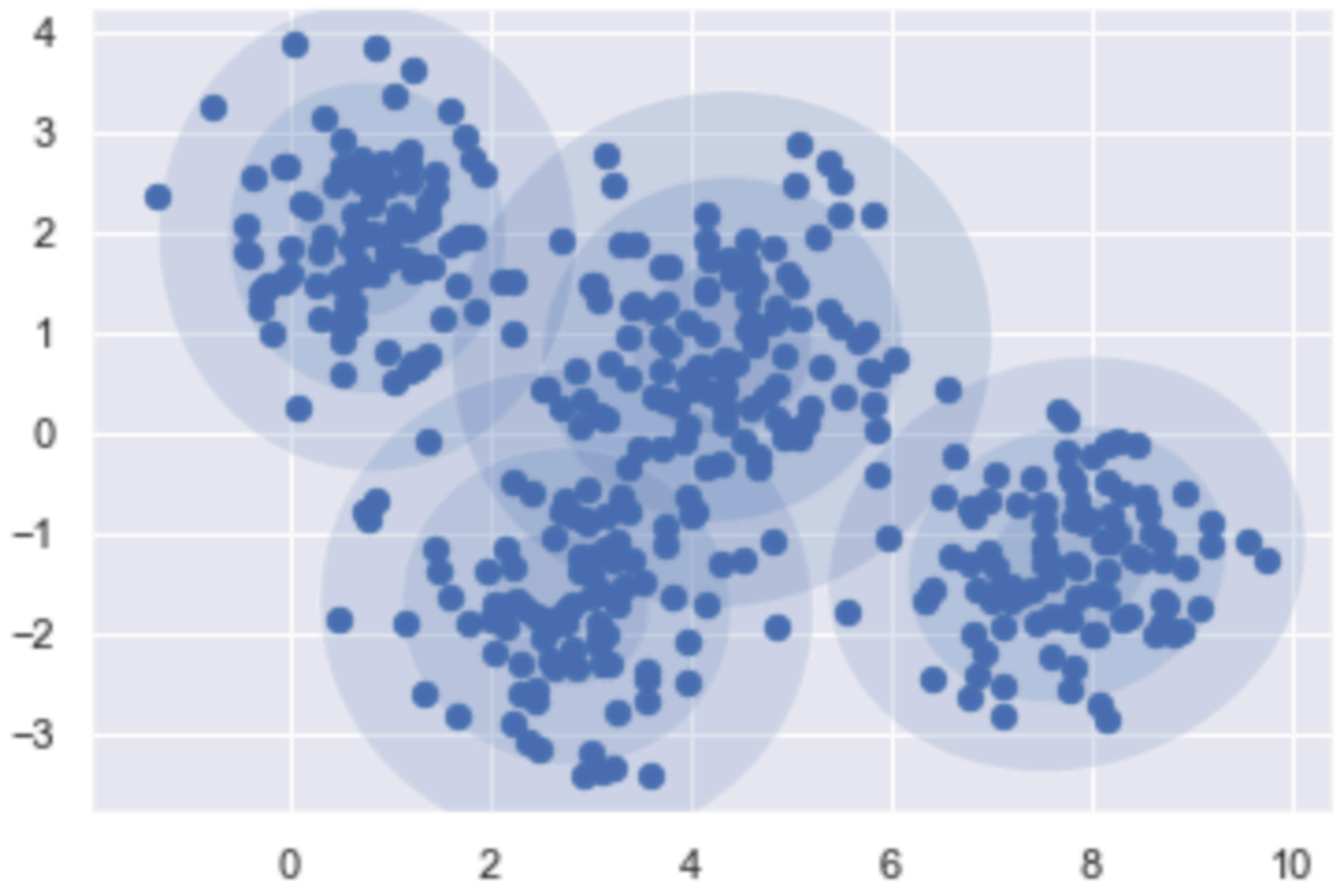
1. Latent Variable Models

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

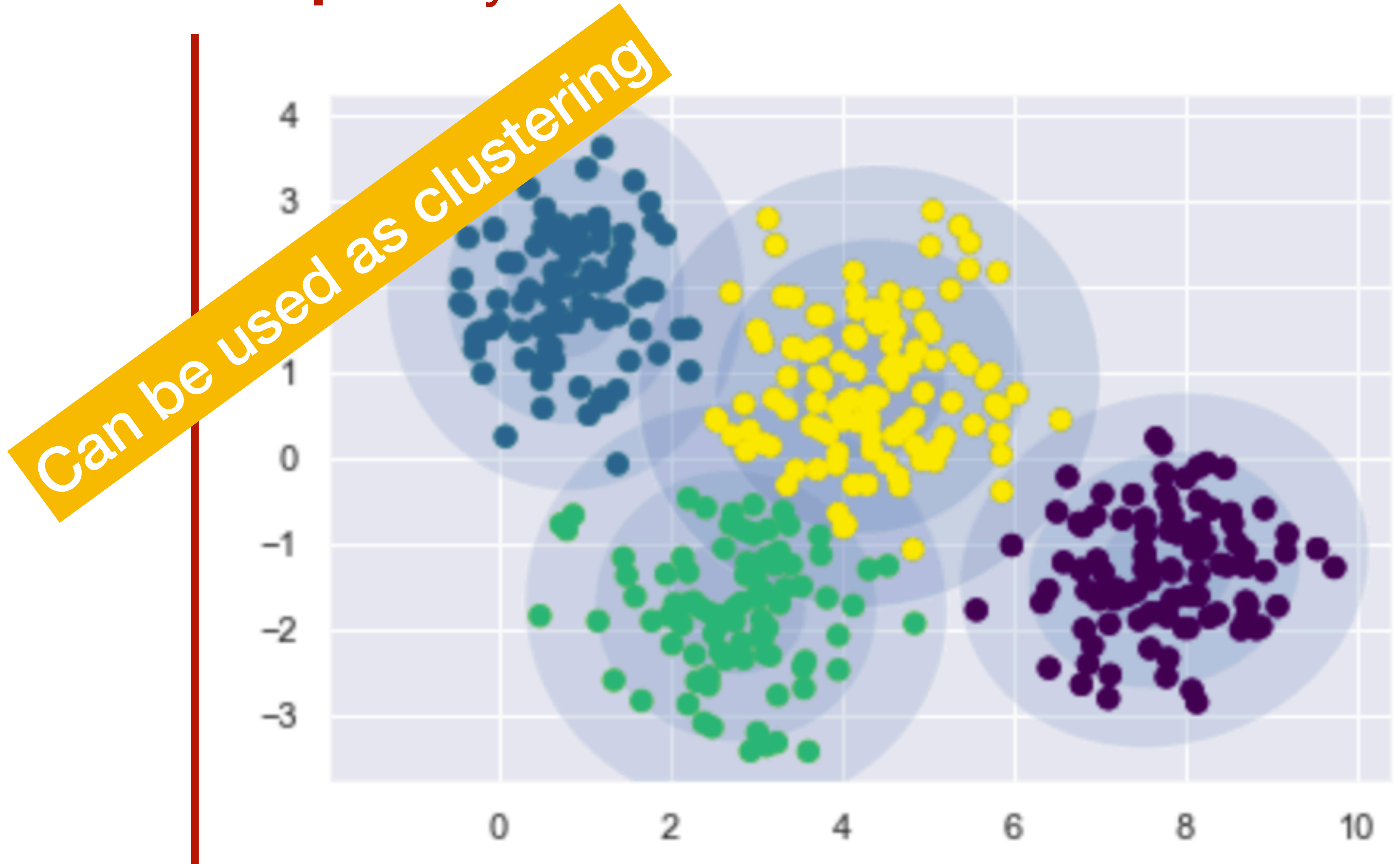
1. Latent Variable Models

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

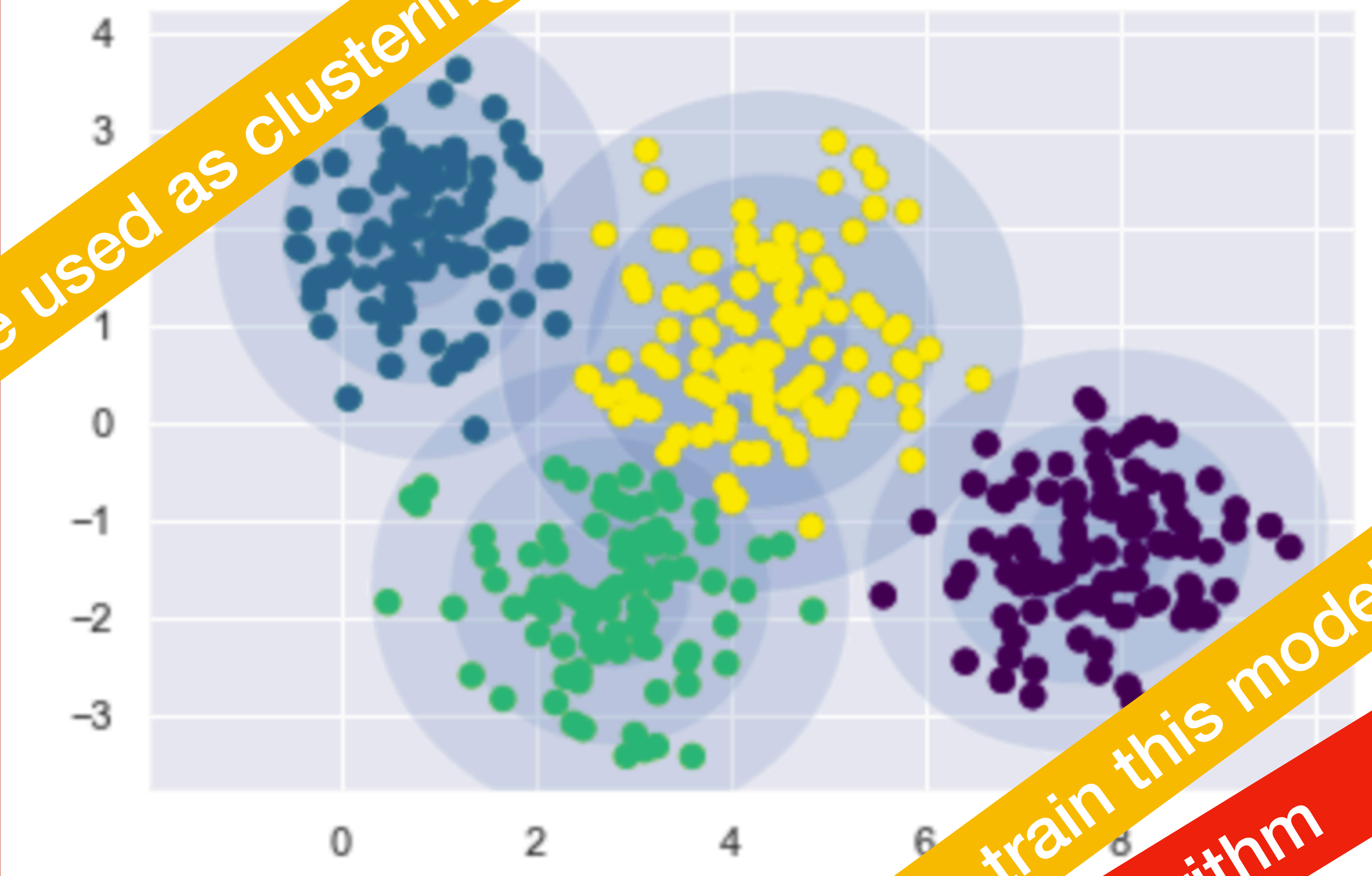
1. Latent Variable Models

Gaussian Mixture Model : 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) !

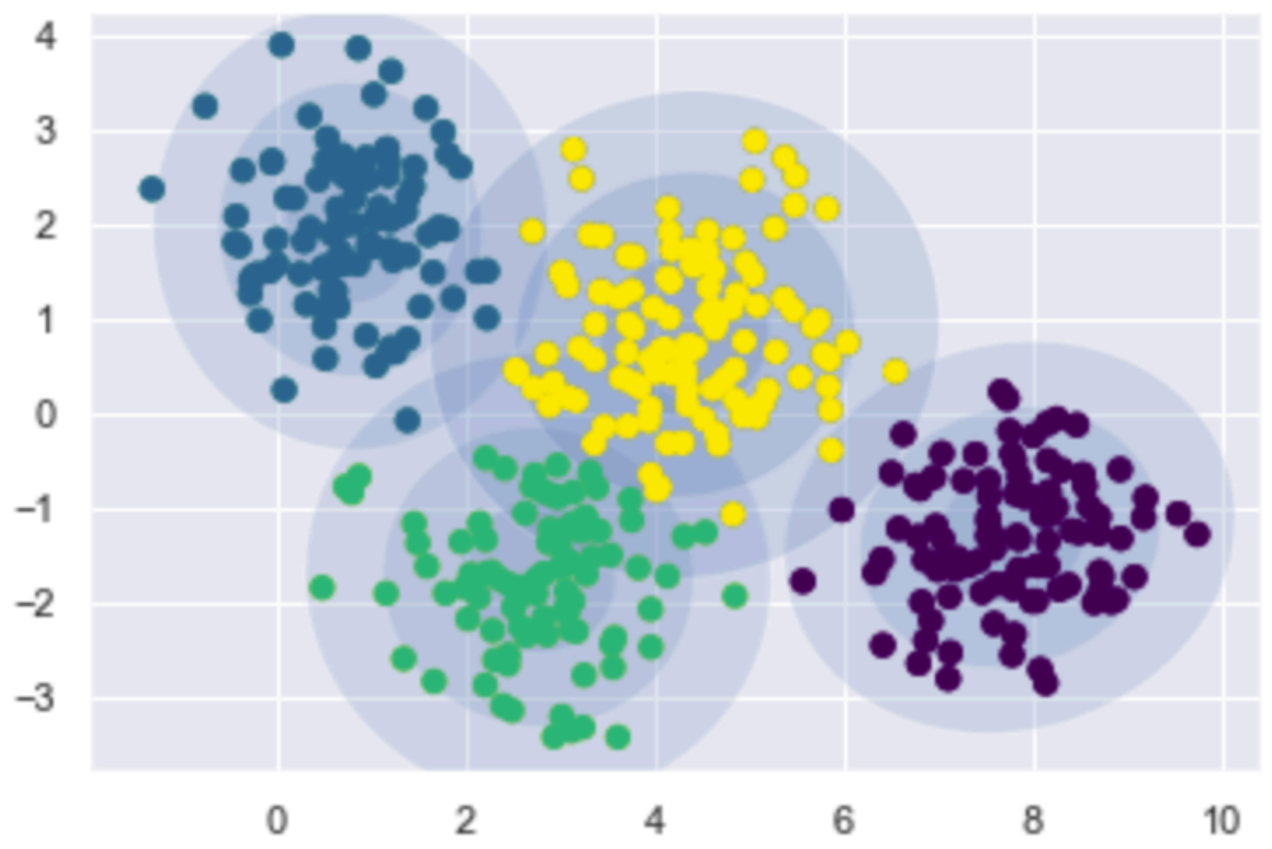
$$\sum_{k=1}^{4} \pi_k \cdot \mathcal{N}(\mu_k, \Sigma_k)$$

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

2 Probabilistic clustering and EM-algorithm

2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model



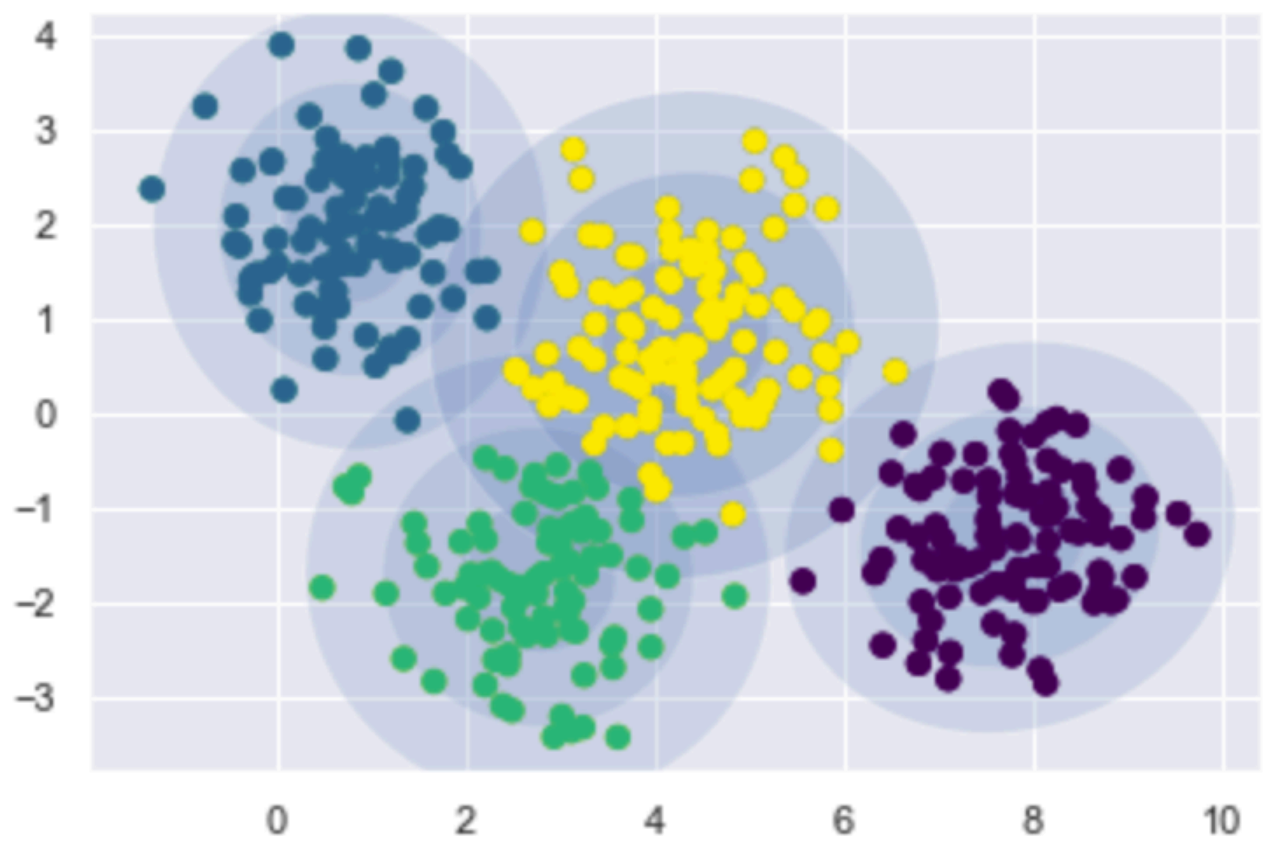
We want to fit a Gaussian Mixture Model !

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

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 :

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

Gaussian
distribution

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

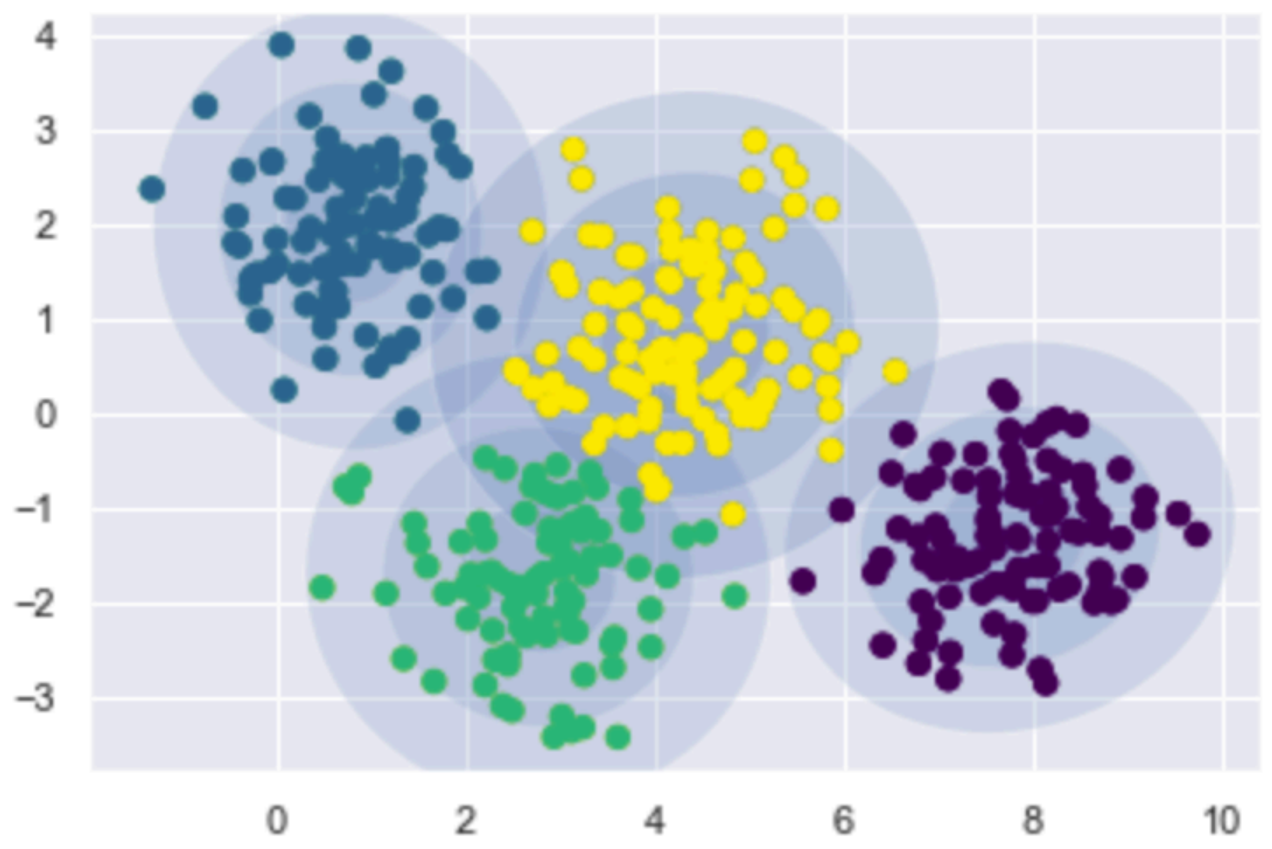
$p(x \,|\, t = k, \theta) =$

$p(x \,|\, \theta) =$

Reminder : a PGM models how an observation is generated

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 :

$$t \longrightarrow x$$

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

x: Gaussian distribution

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

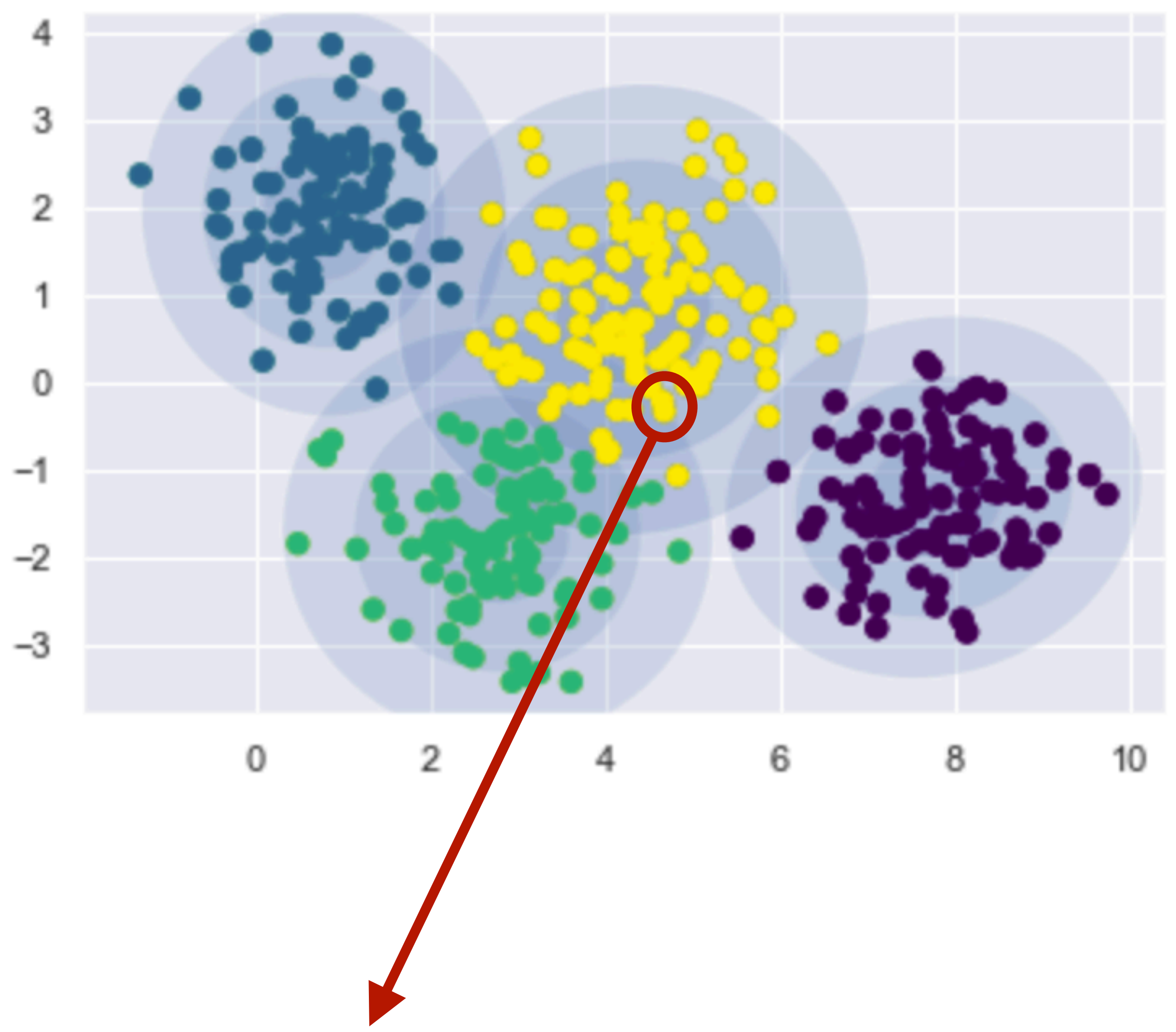
$$p(x \mid t = k, \theta) = \mathcal{N}(x \mid \mu_k, \Sigma_k)$$

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

Reminder : a PGM models how an observation is generated

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

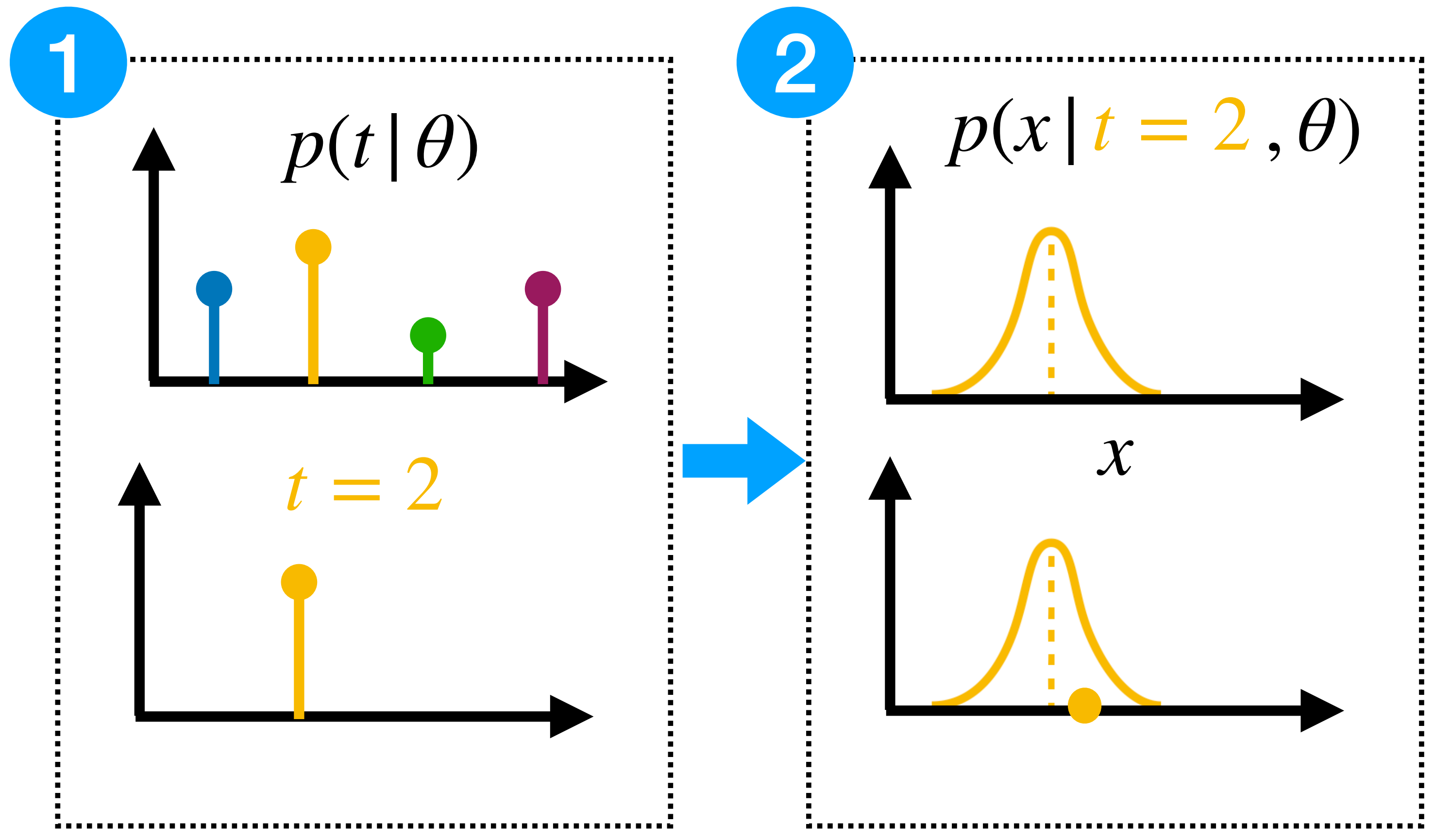
$$p(t=k\,|\,\theta)=\pi_k$$

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

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

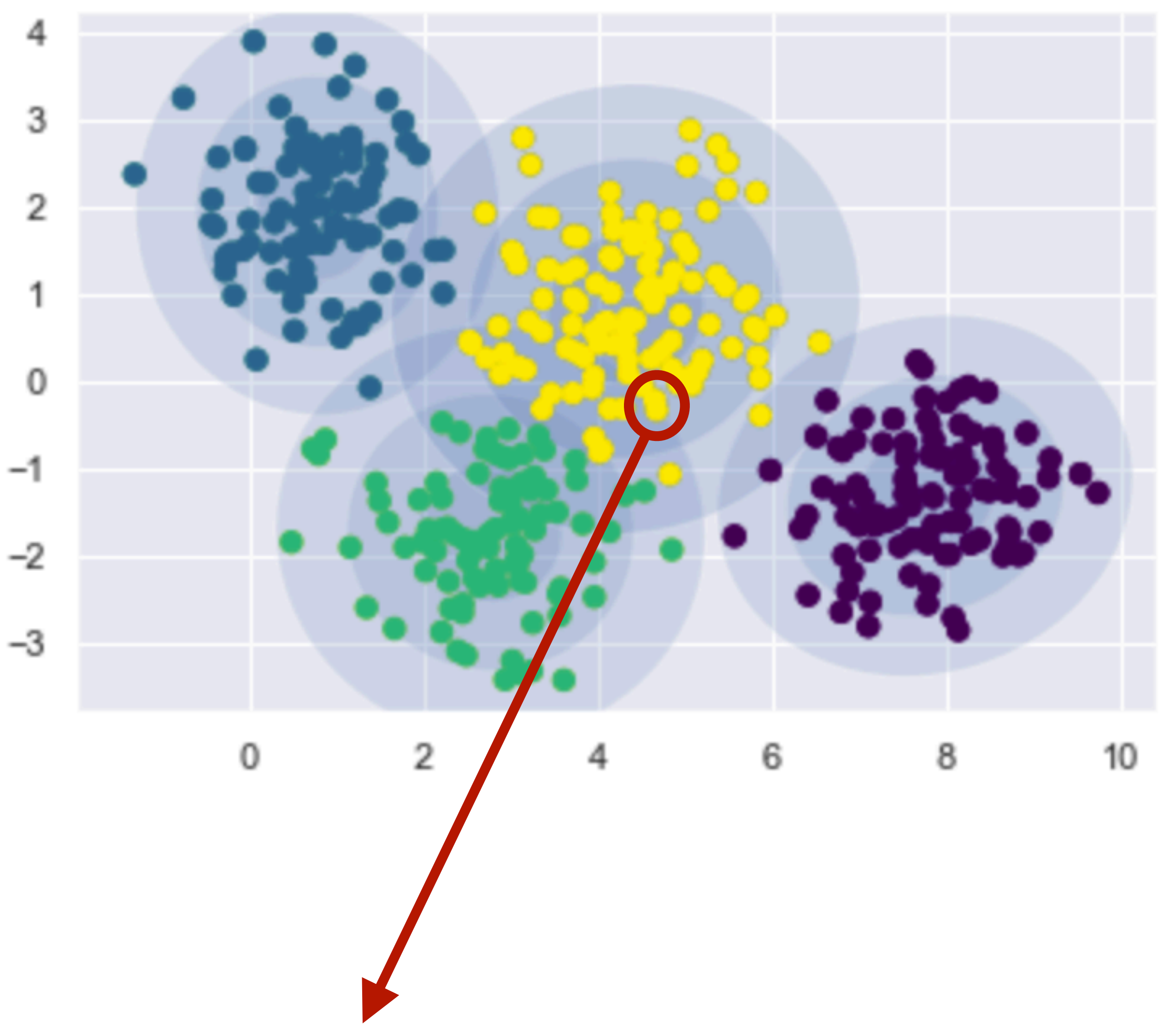
Reminder : a PGM models how an observation is generated

We assume that this x is generated as follows :



2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model

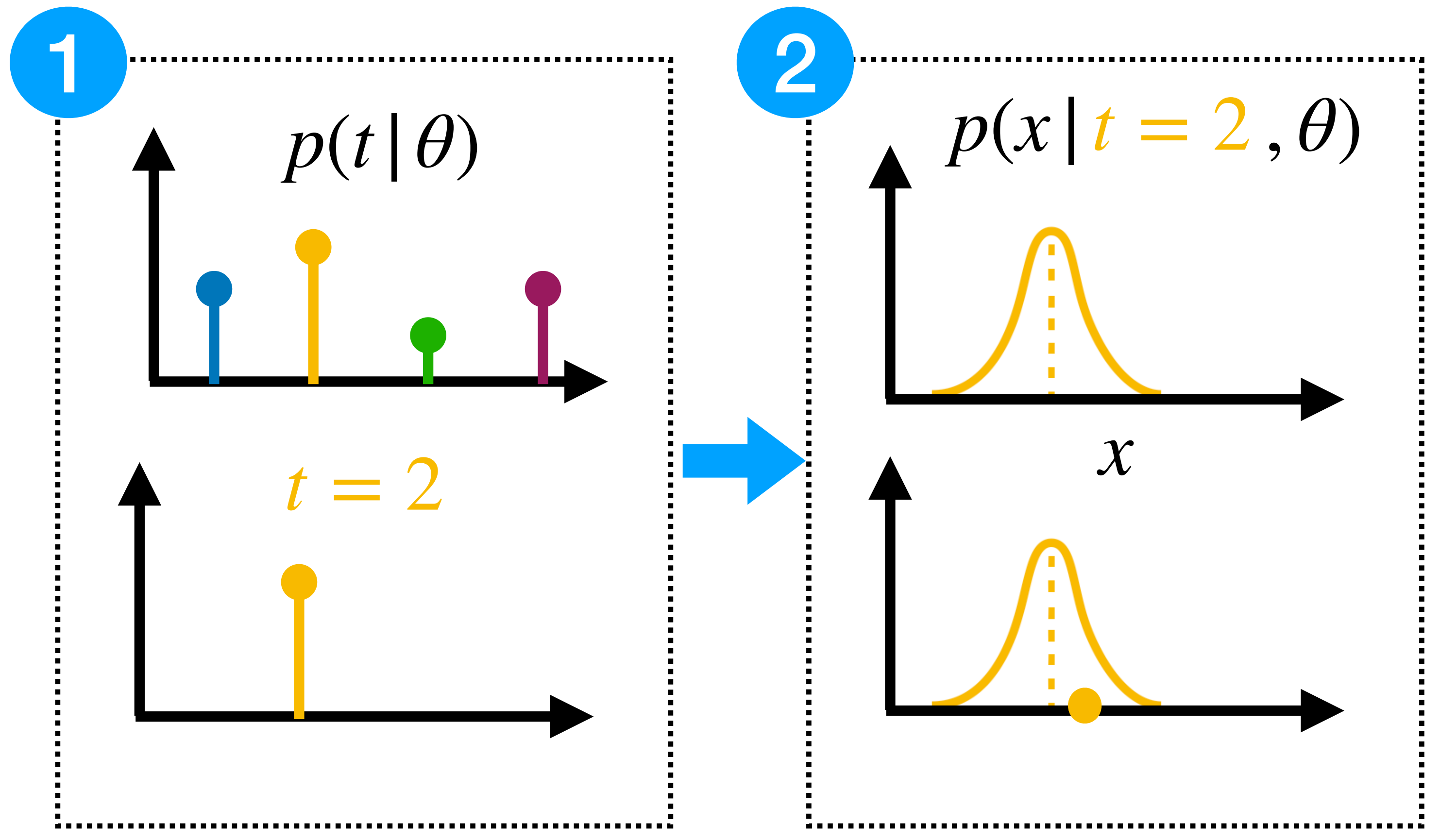


We want to fit a Gaussian Mixture Model !

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

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

We assume that this x is generated as follows :

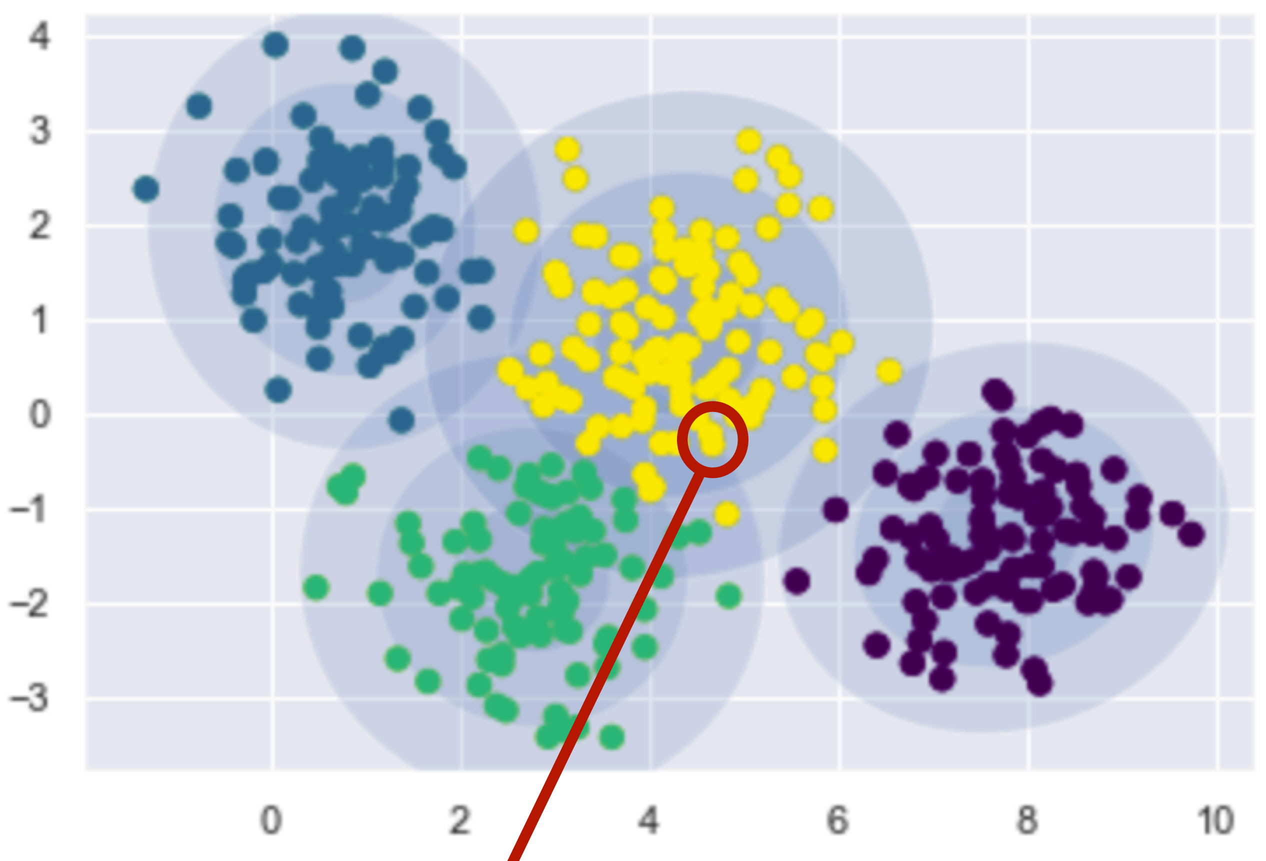


Hard clustering : if we **know the source** of each instances then,

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

2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model

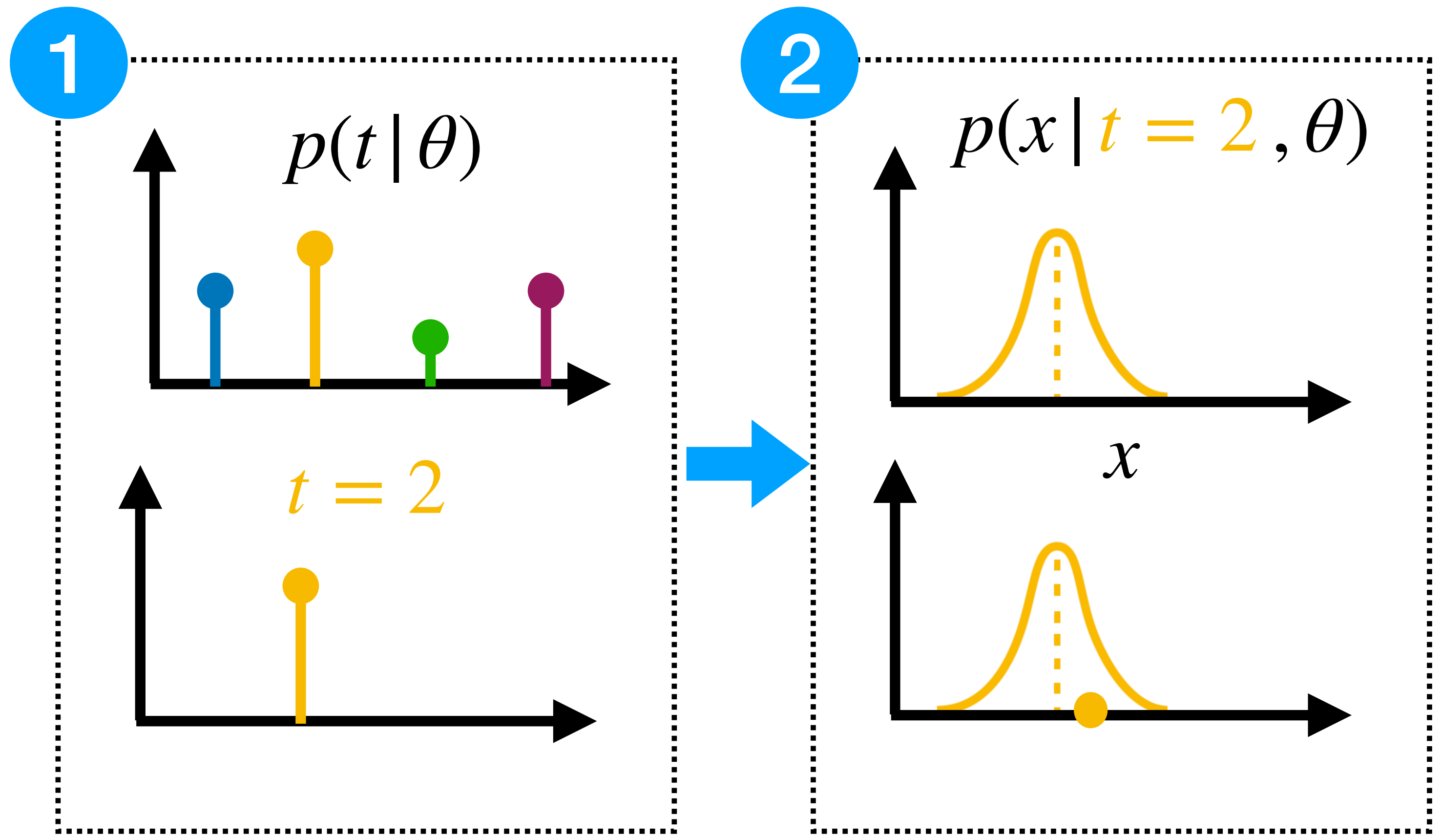


We want to fit a Gaussian Mixture Model !

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

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

We assume that this x is generated as follows :



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

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

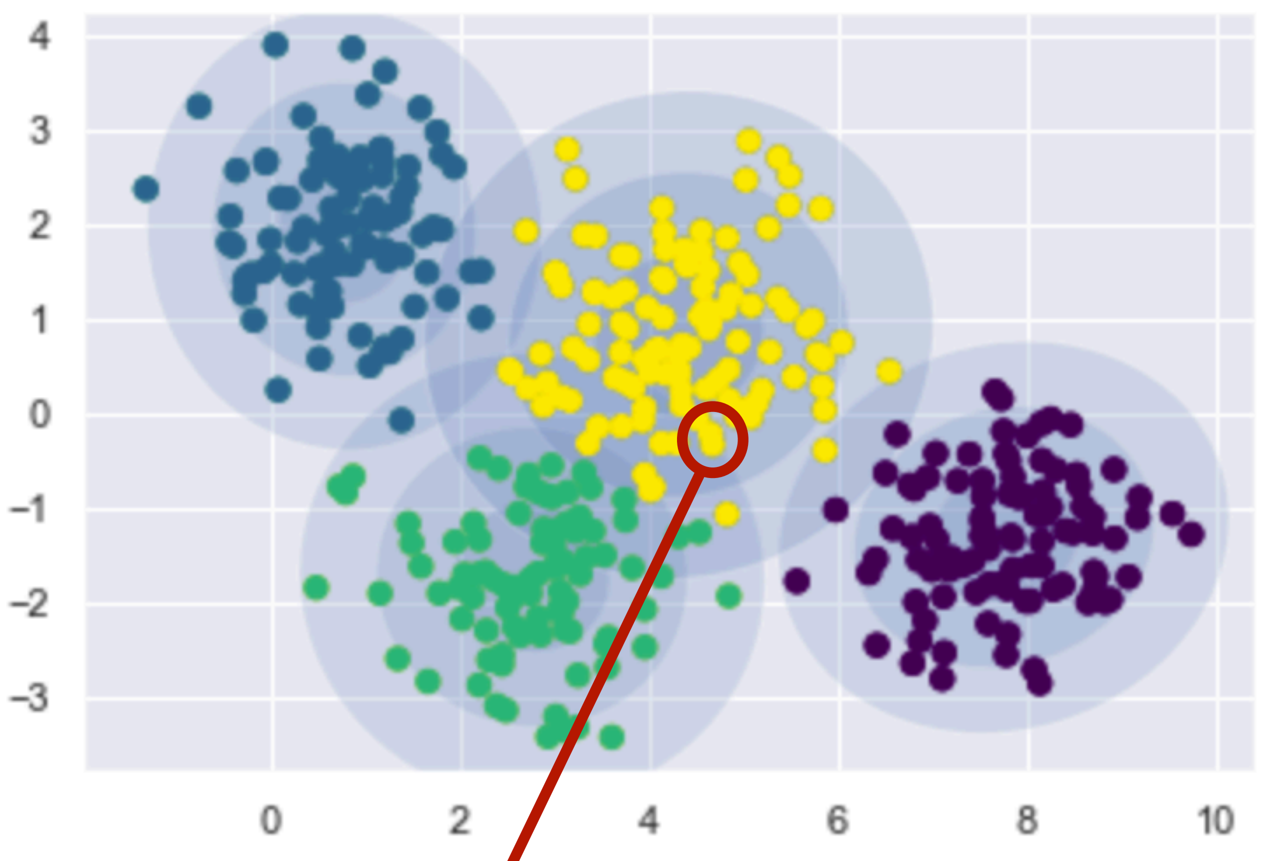
$$\mu_{hard}^{MLE} = \frac{\sum_{i \in \text{cluster 2}} x_i}{\text{Number of points in cluster 2}}$$

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

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

2. Probabilistic clustering

Gaussian Mixture Model as a Latent variable model

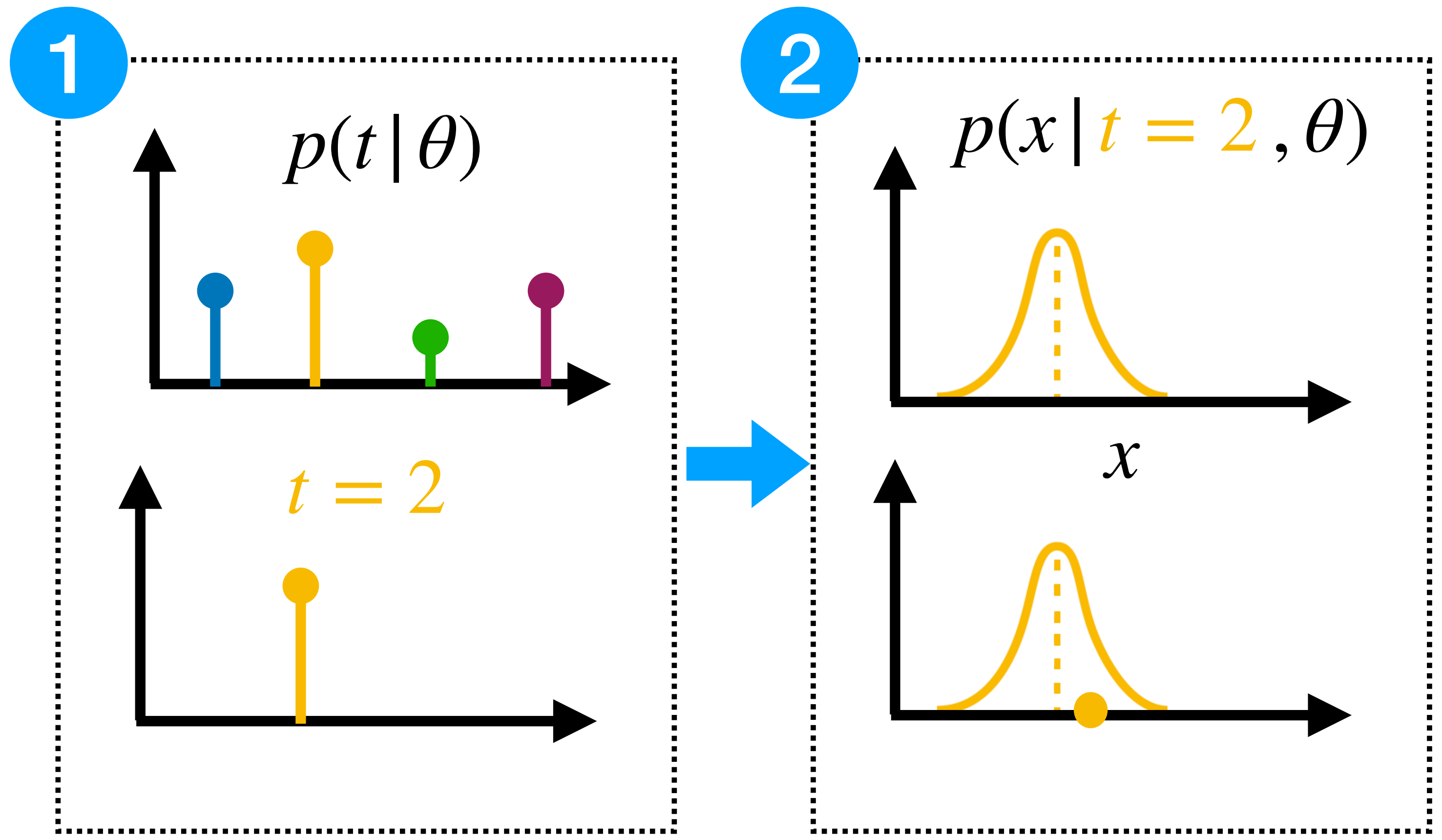


We want to fit a Gaussian Mixture Model !

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

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

We assume that this x is generated as follows :



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,

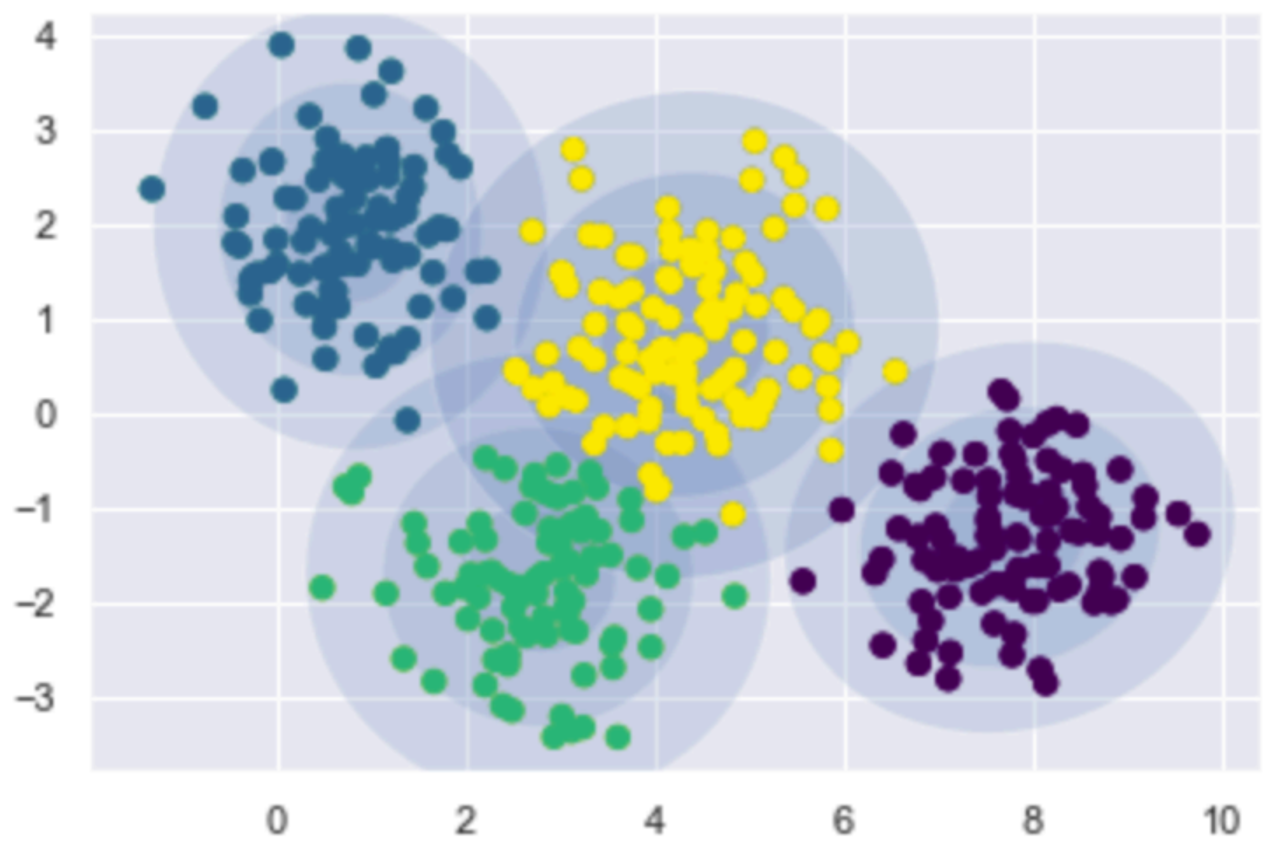
$$p(x\,|\,t=2,\theta)=\mathcal{N}(x\,|\,\mu_{soft}^{MLE},\Sigma_{soft}^{MLE})$$

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

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

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,

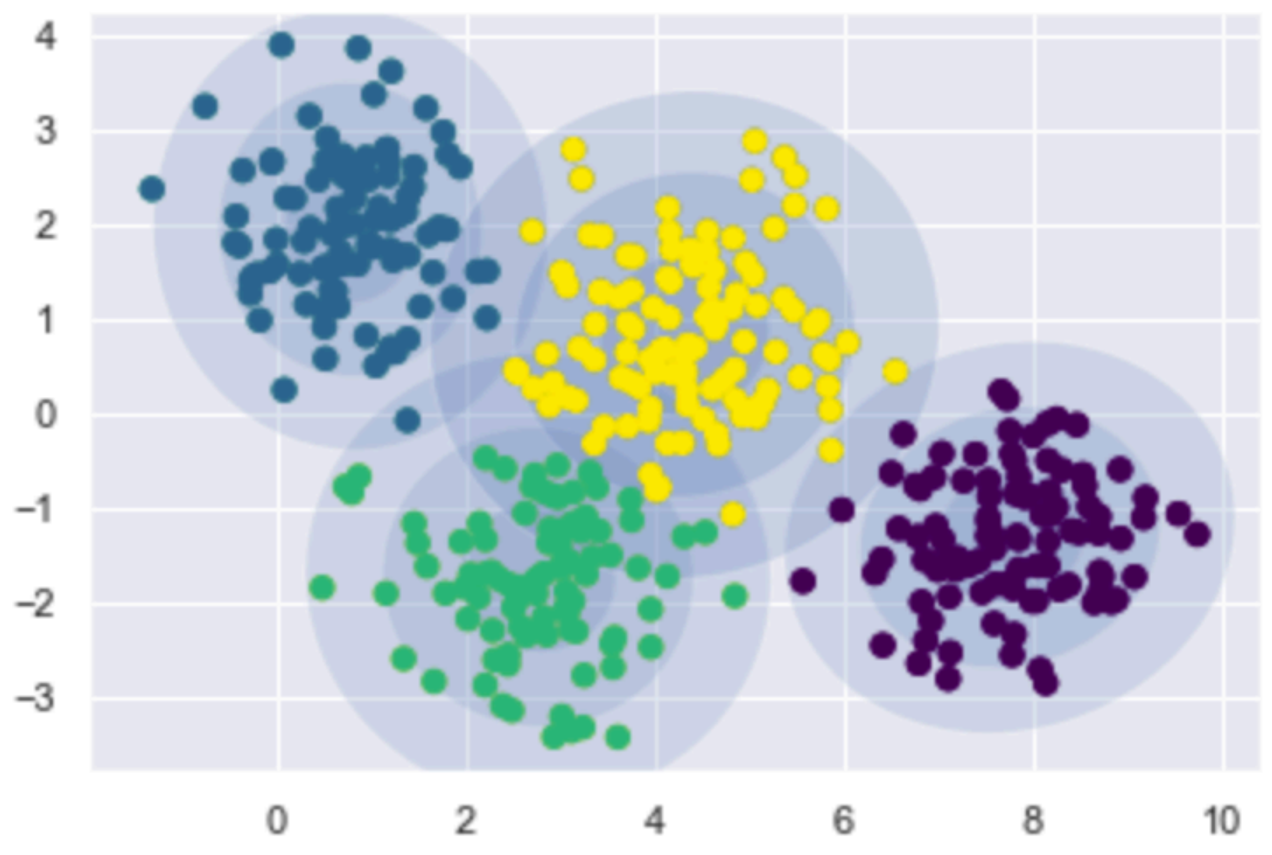
$$p(x \mid t = 2, \theta) = \mathcal{N}(x \mid \mu_{soft}^{MLE}, \Sigma_{soft}^{MLE})$$

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

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

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,

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

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

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

Remarks: If we **know the parameters** of each instances then**,**

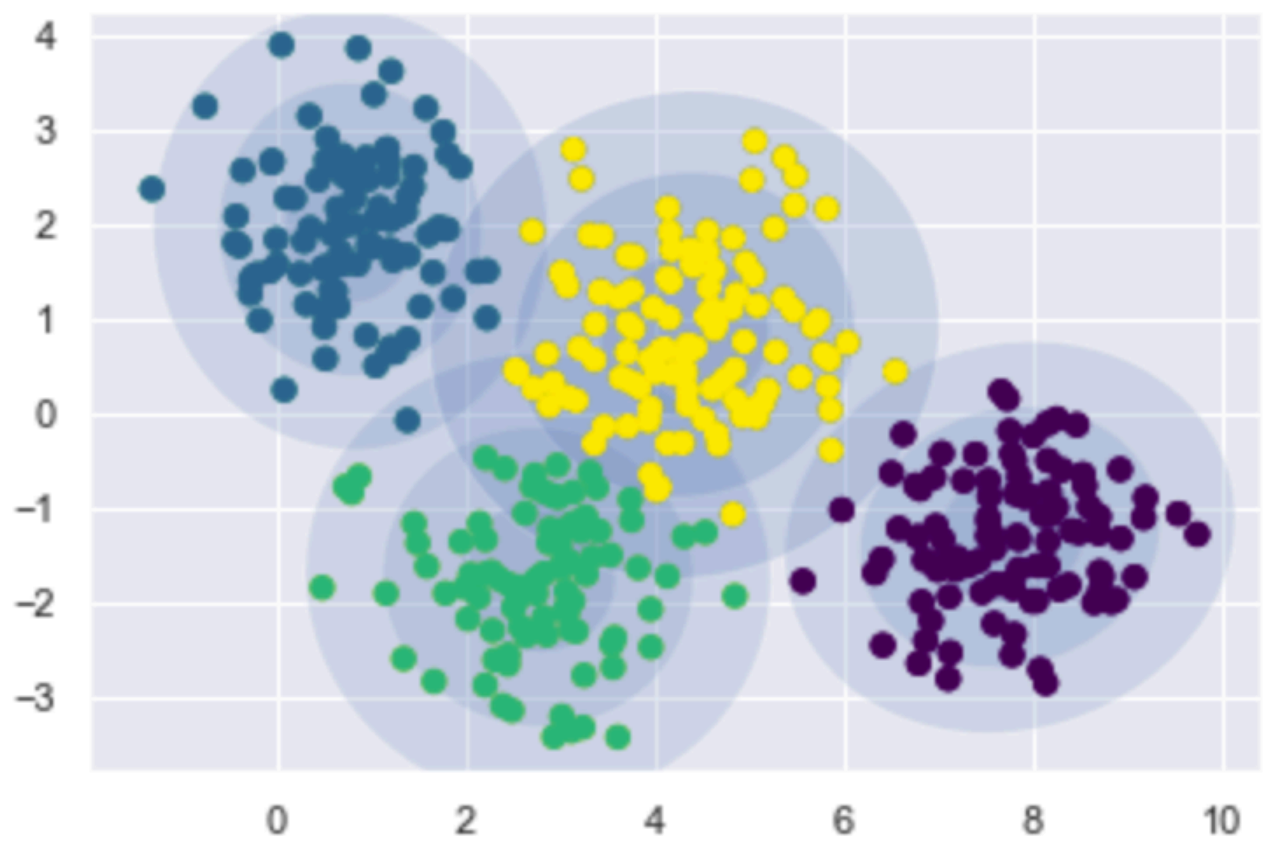
$$p(t = 2 \mid x, \theta) = \frac{p(x \mid t = 2, \theta) \times p(t = 2 \mid \theta)}{\text{Const}}$$

We are now in the following situation :

- **ESTIMATION:**
  If we **knew the parameters**, we could compute the posteriors
- **MAXIMIZATION:**
  If we **knew the posteriors/ sources**, we could easily compute the parameters

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,

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

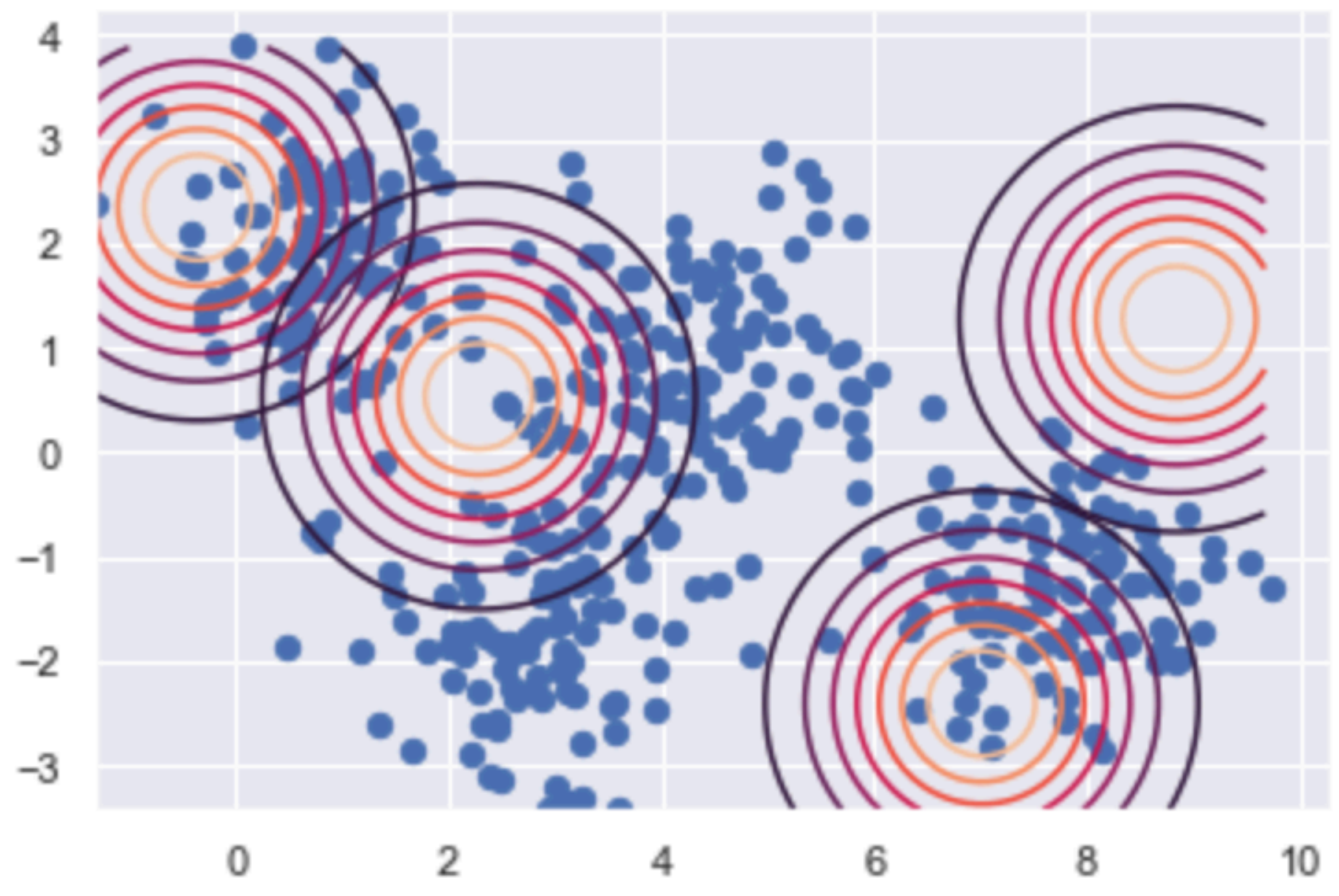
$$\mu_{soft}^{MLE} = \frac{\sum_i \; p(t = 2 \,|\, x, \theta)\; x_i}{\sum_i p(\; t = 2 \,|\, x, \theta)}$$

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

Remarks: If we **know the parameters** of each instances then**,**

$$p(t = 2 \,|\, x, \theta) = \frac{p(x \,|\, t = 2, \theta) \times p(t = 2 \,|\, \theta)}{\text{Const}}$$

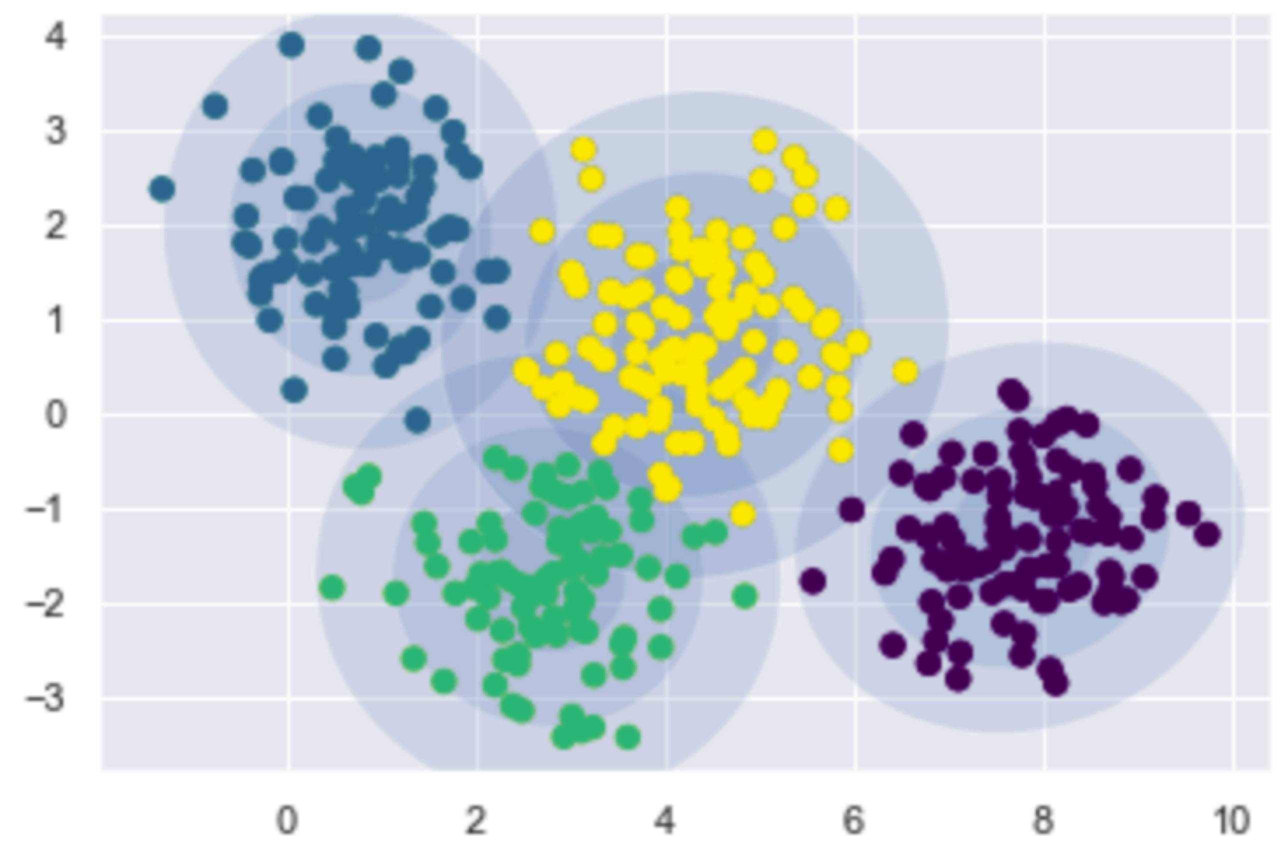
INITIALISATION : first estimation



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

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,

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

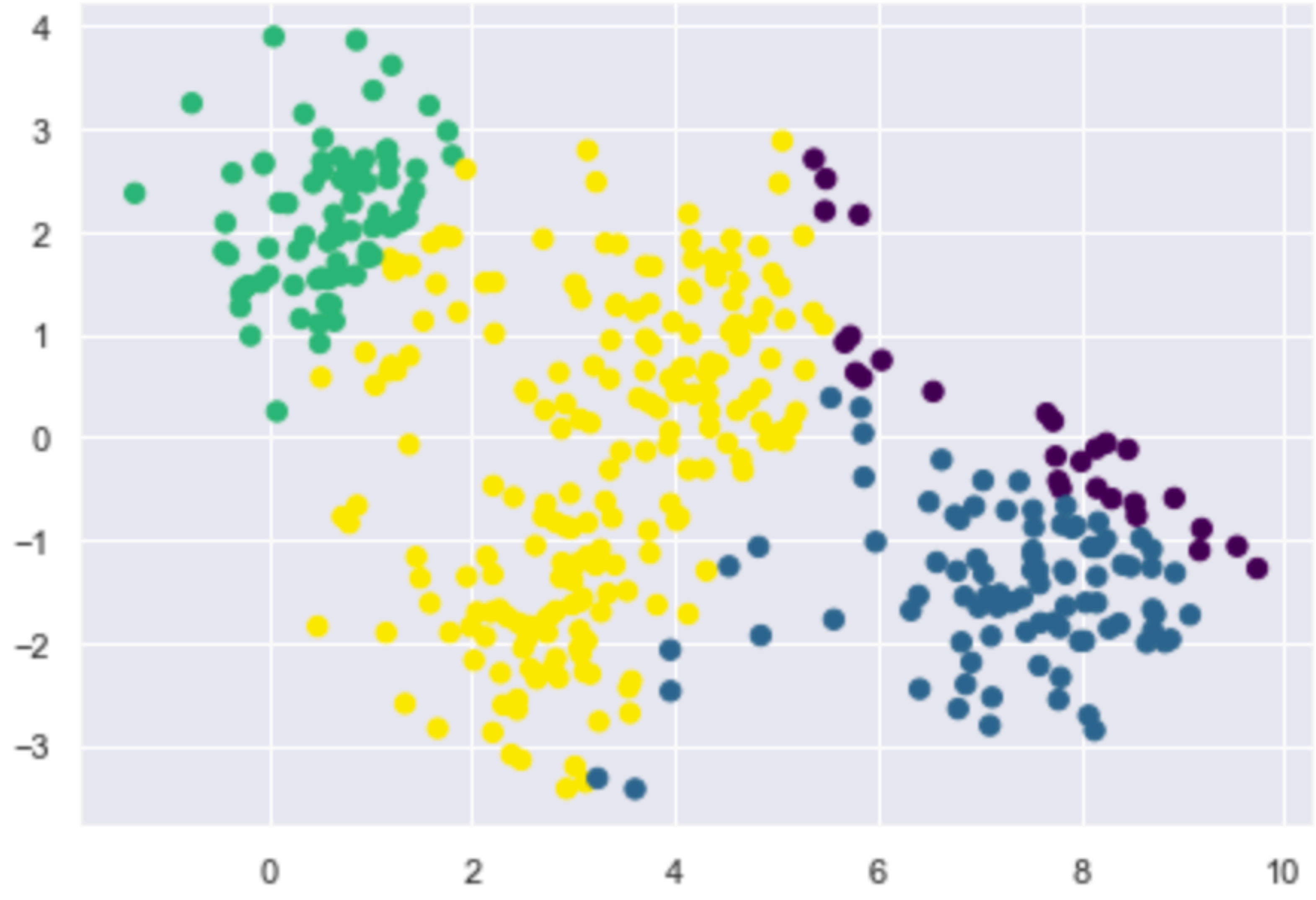
$$\mu_{soft}^{MLE} = \frac{\sum_i \; p(t=2 \,|\, x, \theta)\; x_i}{\sum_i p(\; t=2 \,|\, x, \theta)}$$

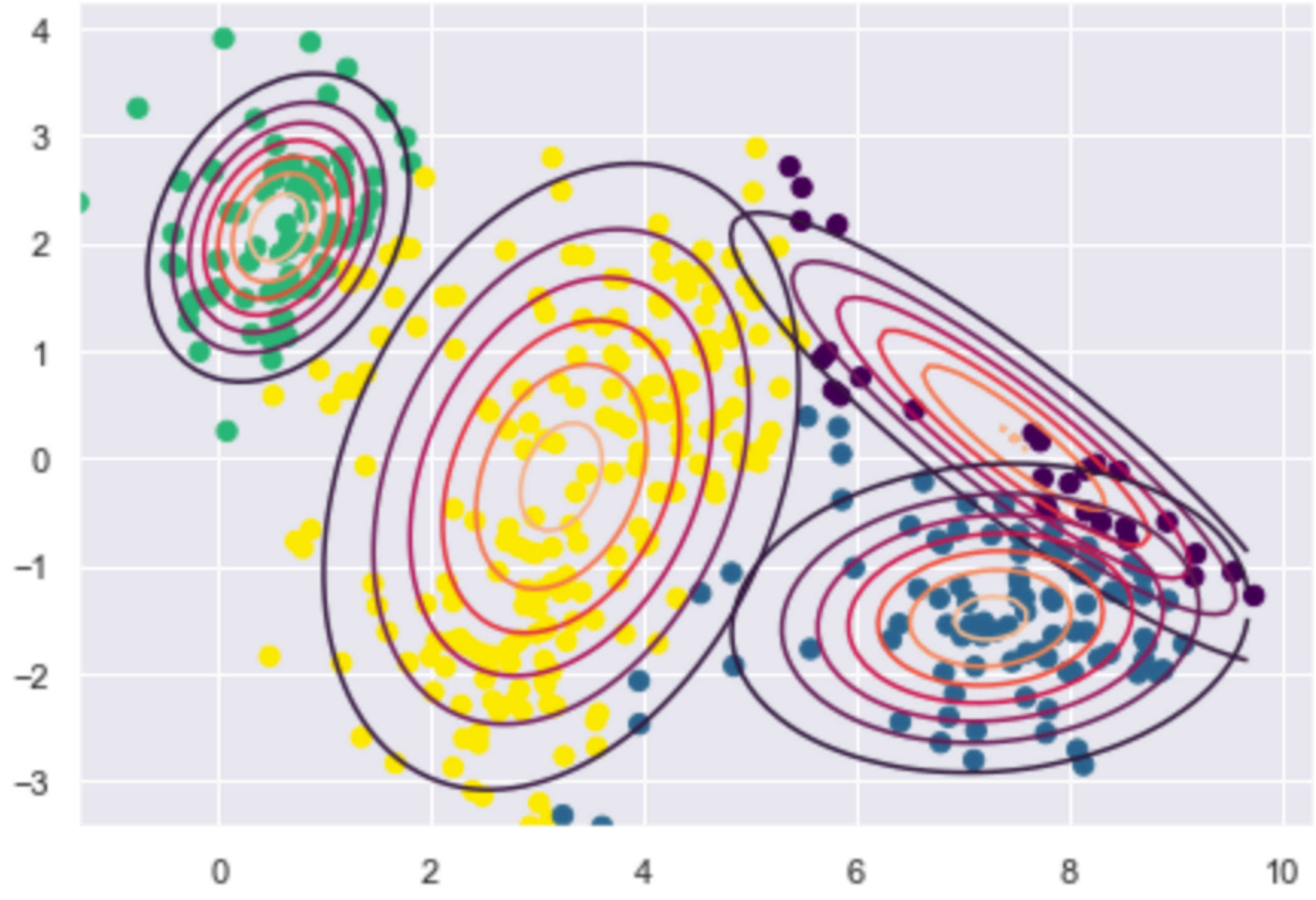
$$\Sigma_{soft}^{MLE} = \frac{\sum_i \; p(t=2 \,|\, x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(\; t=2 \,|\, x_i, \theta)}$$

Remarks: If we **know the parameters** of each instances then**,**

$$p(t=2 \,|\, x, \theta) = \frac{p(x \,|\, t=2, \theta) \times p(t=2 \,|\, \theta)}{\text{Const}}$$

STEP 1



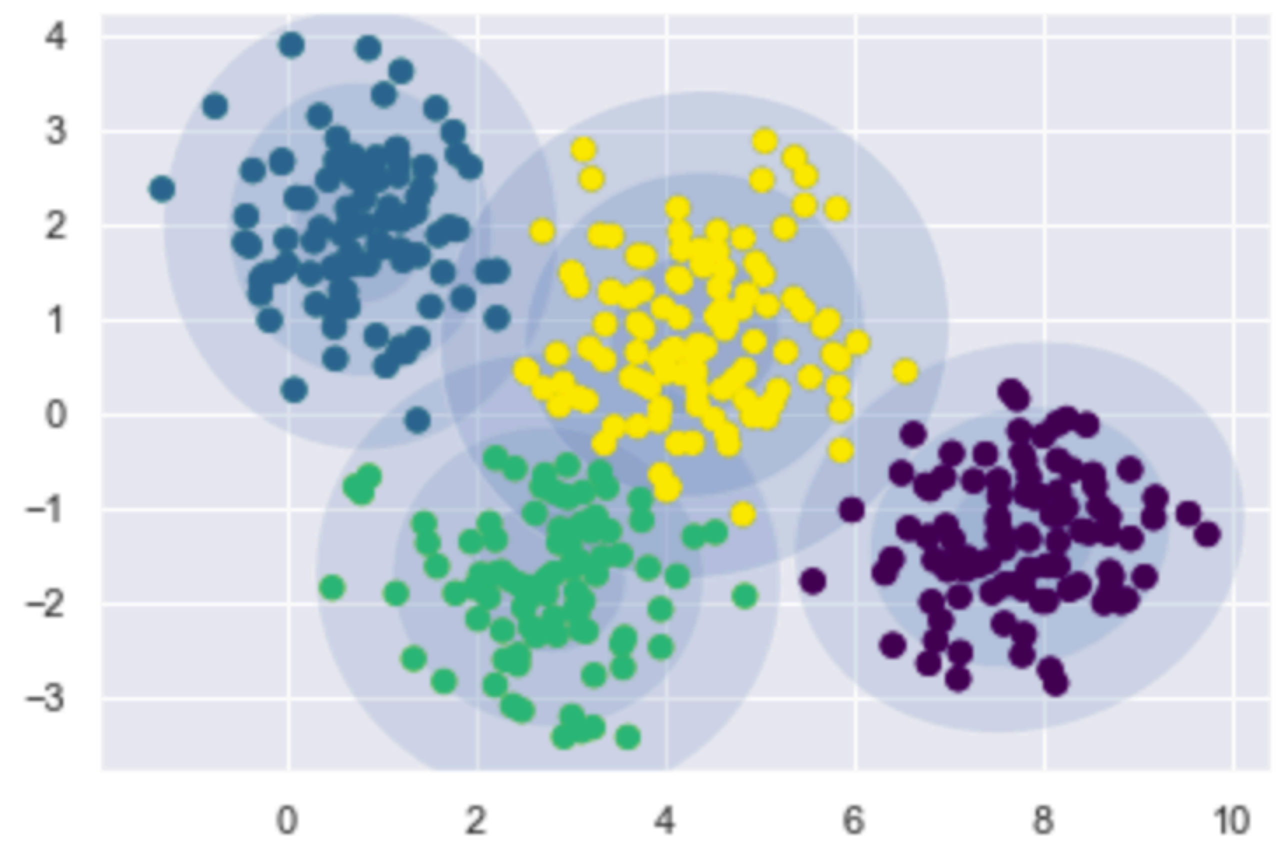


We are now in the following situation :

- **ESTIMATION:**
  If we **knew the parameters**, we could compute the posteriors
- **MAXIMIZATION:**
  If we **knew the posteriors/ sources**, we could easily compute the parameters

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,

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

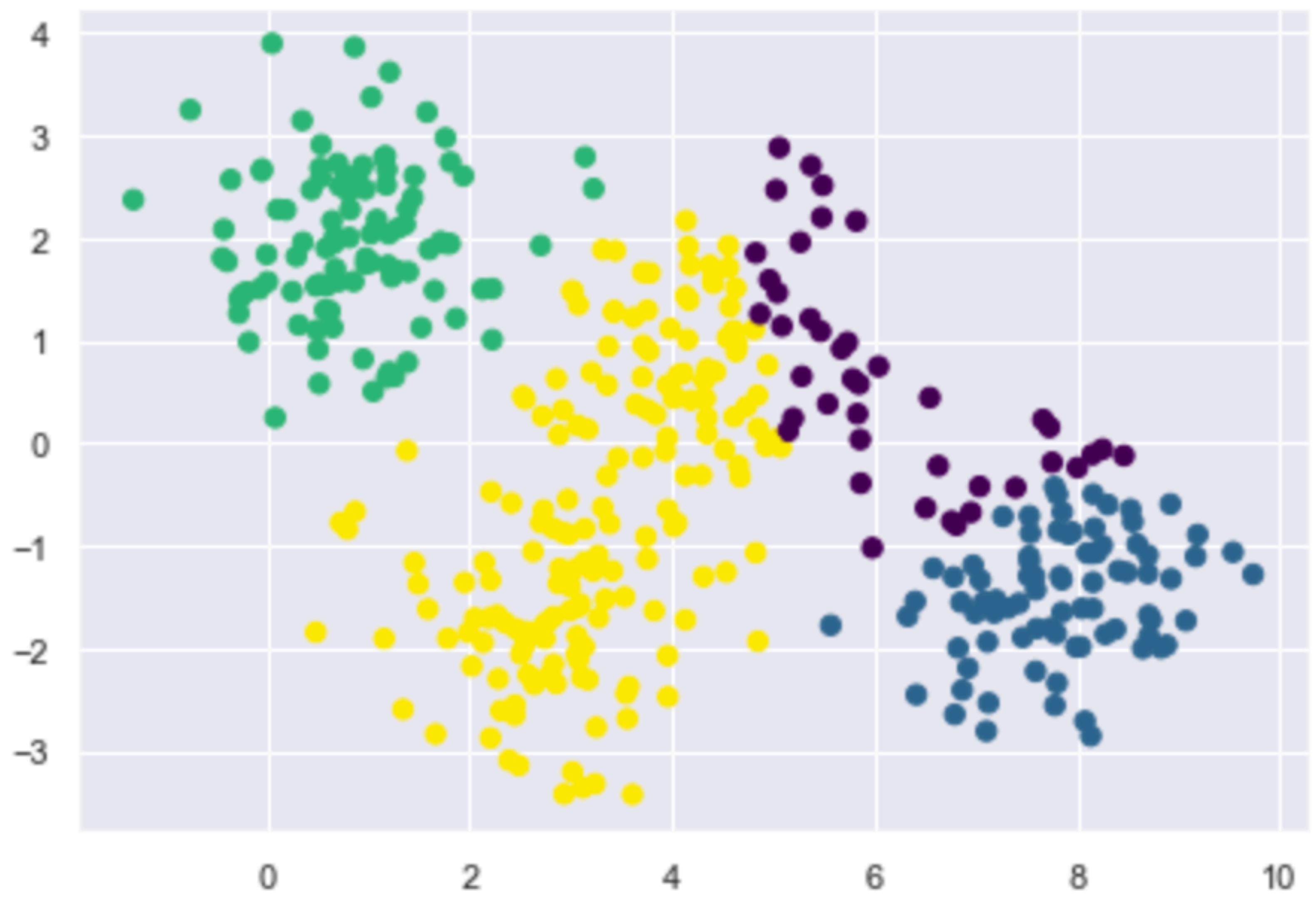
$$\mu_{soft}^{MLE} = \frac{\sum_i \; p(t = 2 \mid x, \theta)\; x_i}{\sum_i p(\; t = 2 \mid x, \theta)}$$

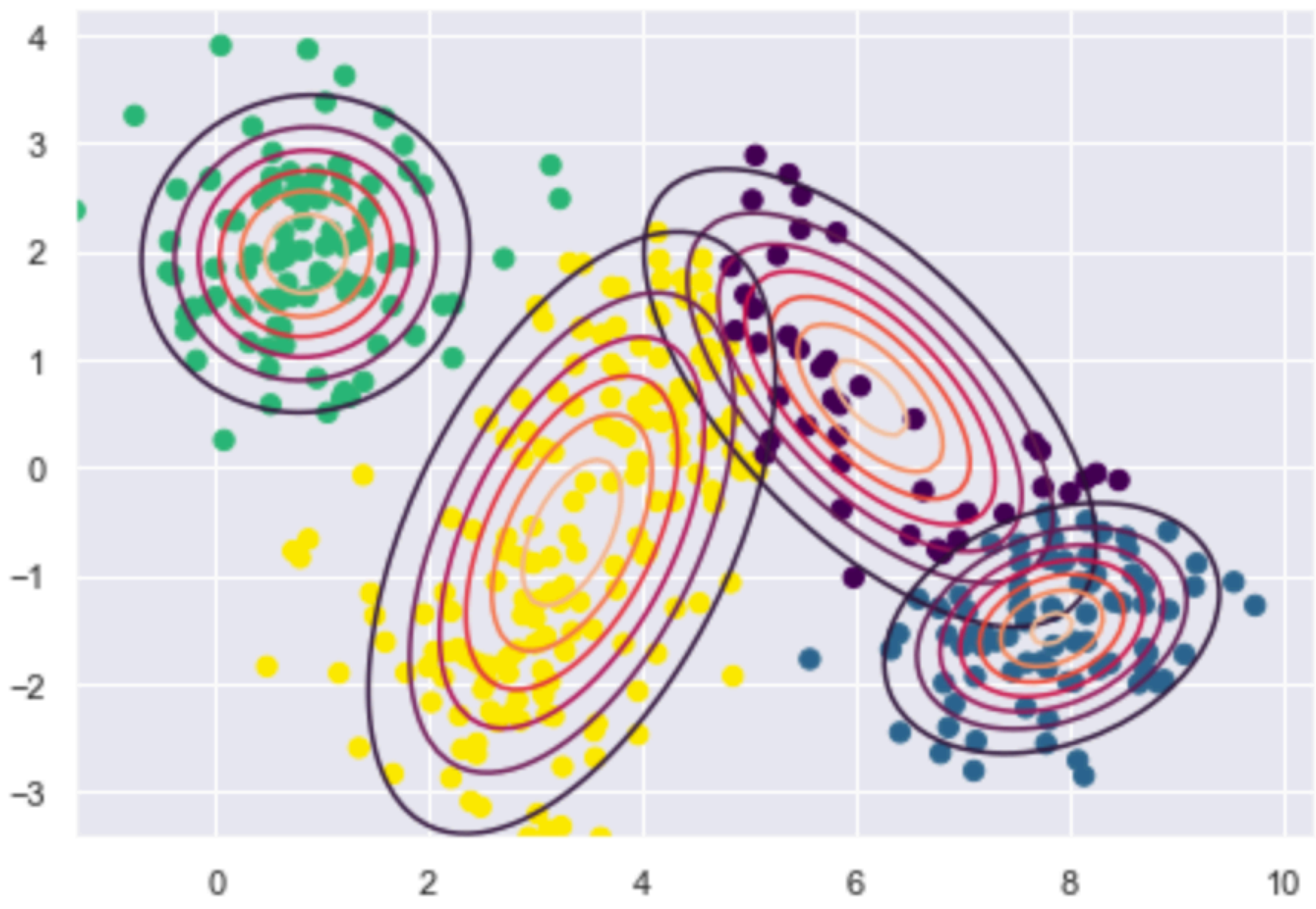
$$\Sigma_{soft}^{MLE} = \frac{\sum_i \; p(t = 2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(\; t = 2 \mid x_i, \theta)}$$

Remarks: If we **know the parameters** of each instances then**,**

$$p(t = 2 \mid x, \theta) = \frac{p(x \mid t = 2, \theta) \times p(t = 2 \mid \theta)}{\text{Const}}$$

STEP 2



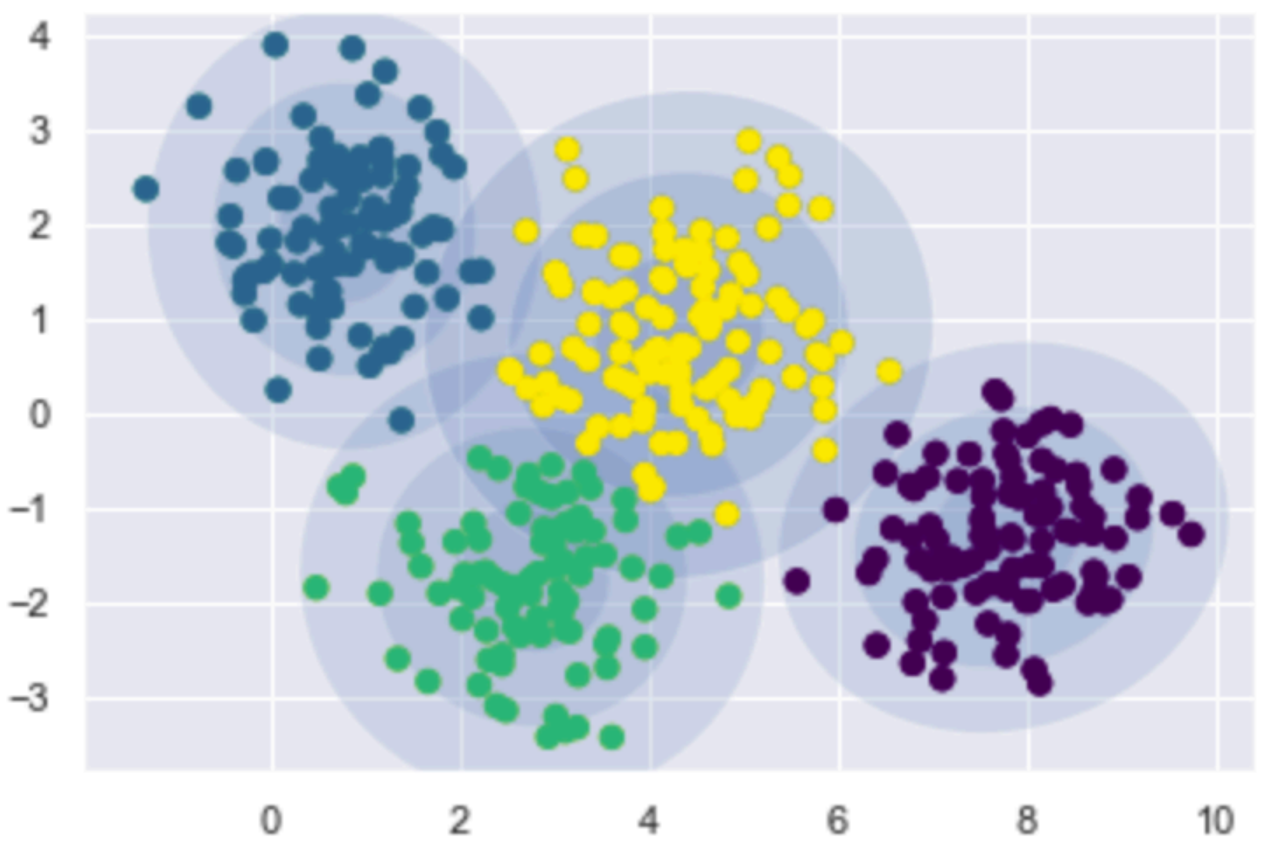


We are now in the following situation :

- **ESTIMATION:**
  If we **knew the parameters**, we could compute the posteriors
- **MAXIMIZATION:**
  If we **knew the posteriors/ sources**, we could easily compute the parameters

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,

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

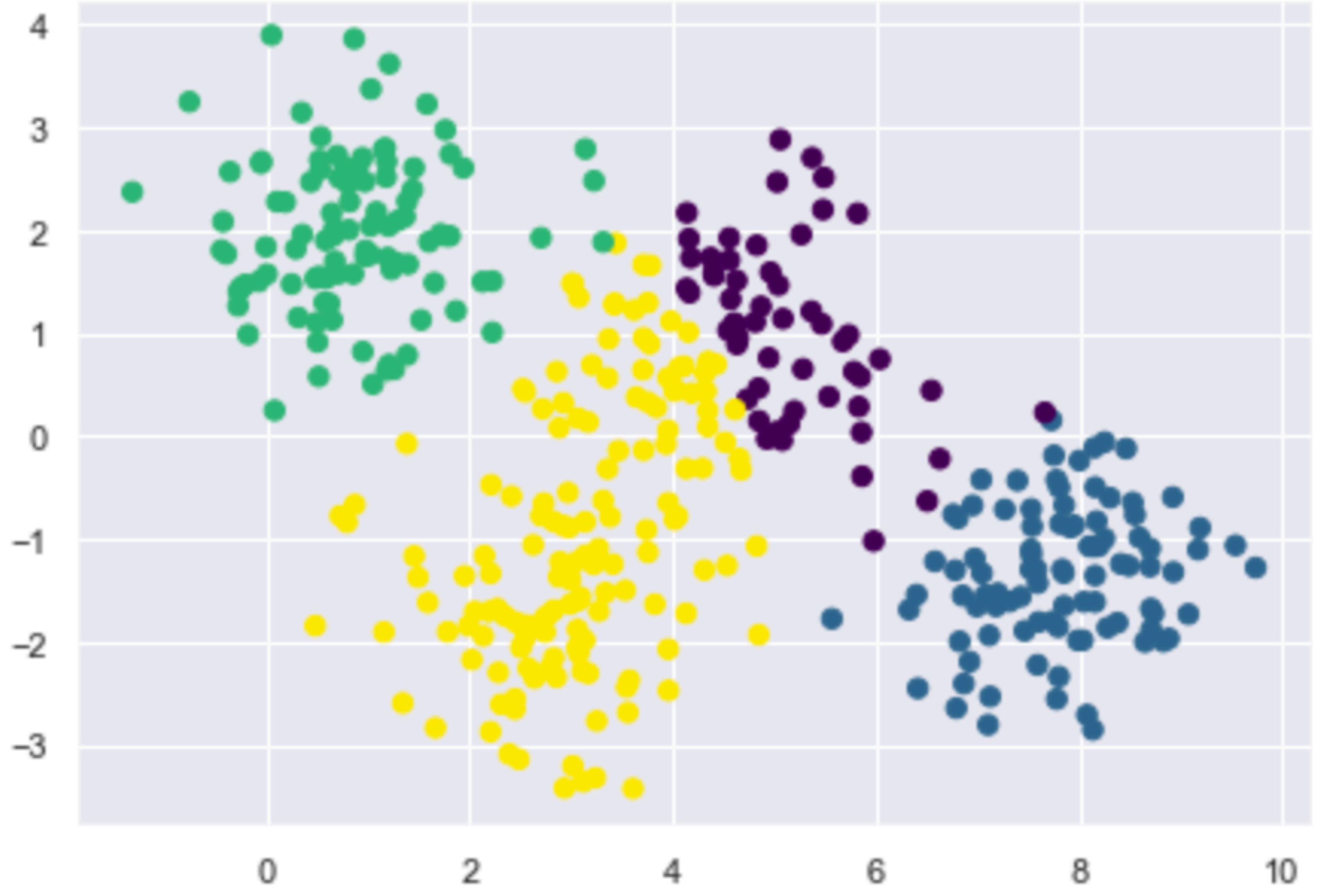
$$\mu_{soft}^{MLE} = \frac{\sum_i \; p(t=2 \,|\, x, \theta)\; x_i}{\sum_i p(\; t=2 \,|\, x, \theta)}$$

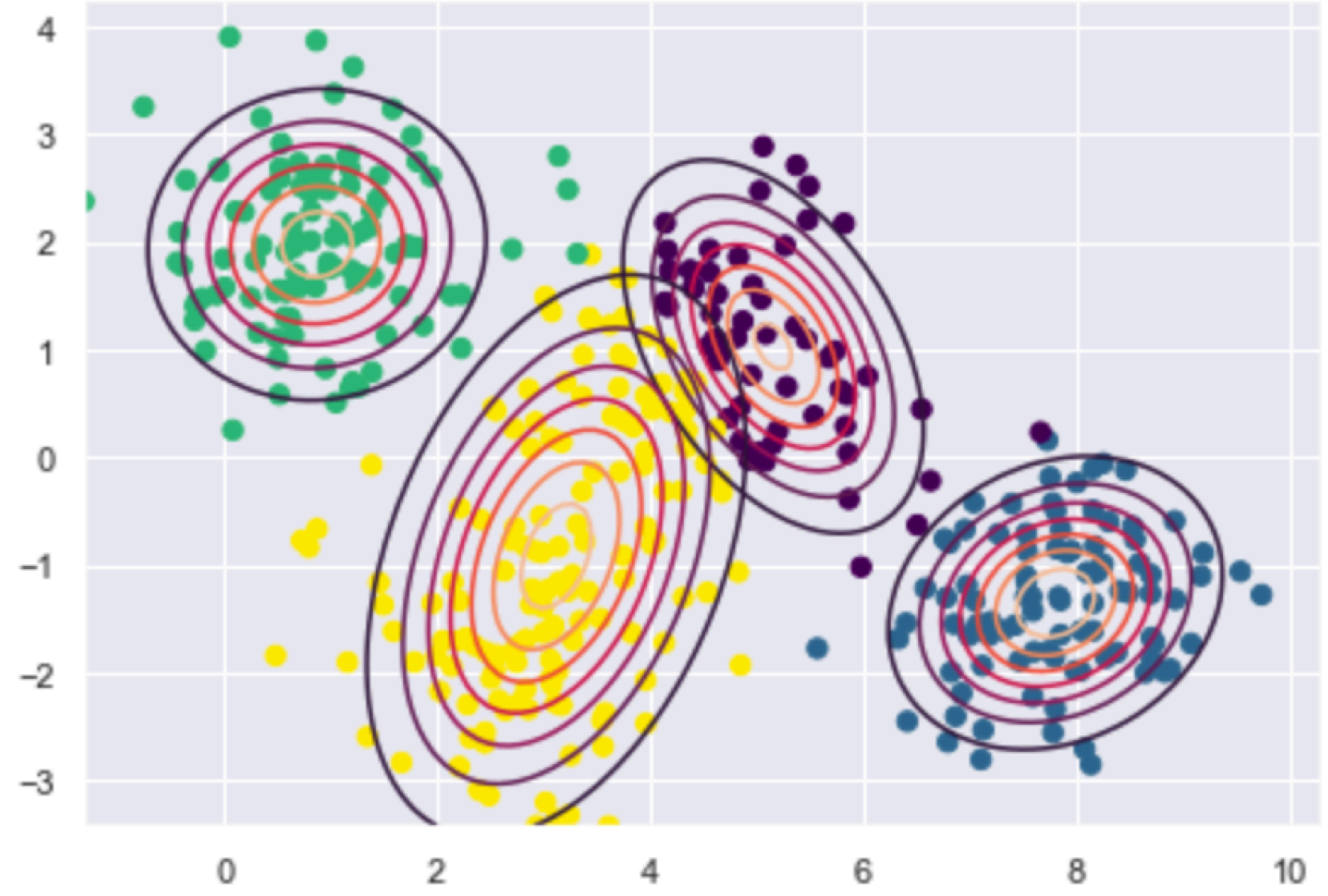
$$\Sigma_{soft}^{MLE} = \frac{\sum_i \; p(t=2 \,|\, x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(\; t=2 \,|\, x_i, \theta)}$$

Remarks: If we **know the parameters** of each instances then**,**

$$p(t=2 \,|\, x, \theta) = \frac{p(x \,|\, t=2, \theta) \times p(t=2 \,|\, \theta)}{\text{Const}}$$

STEP 3



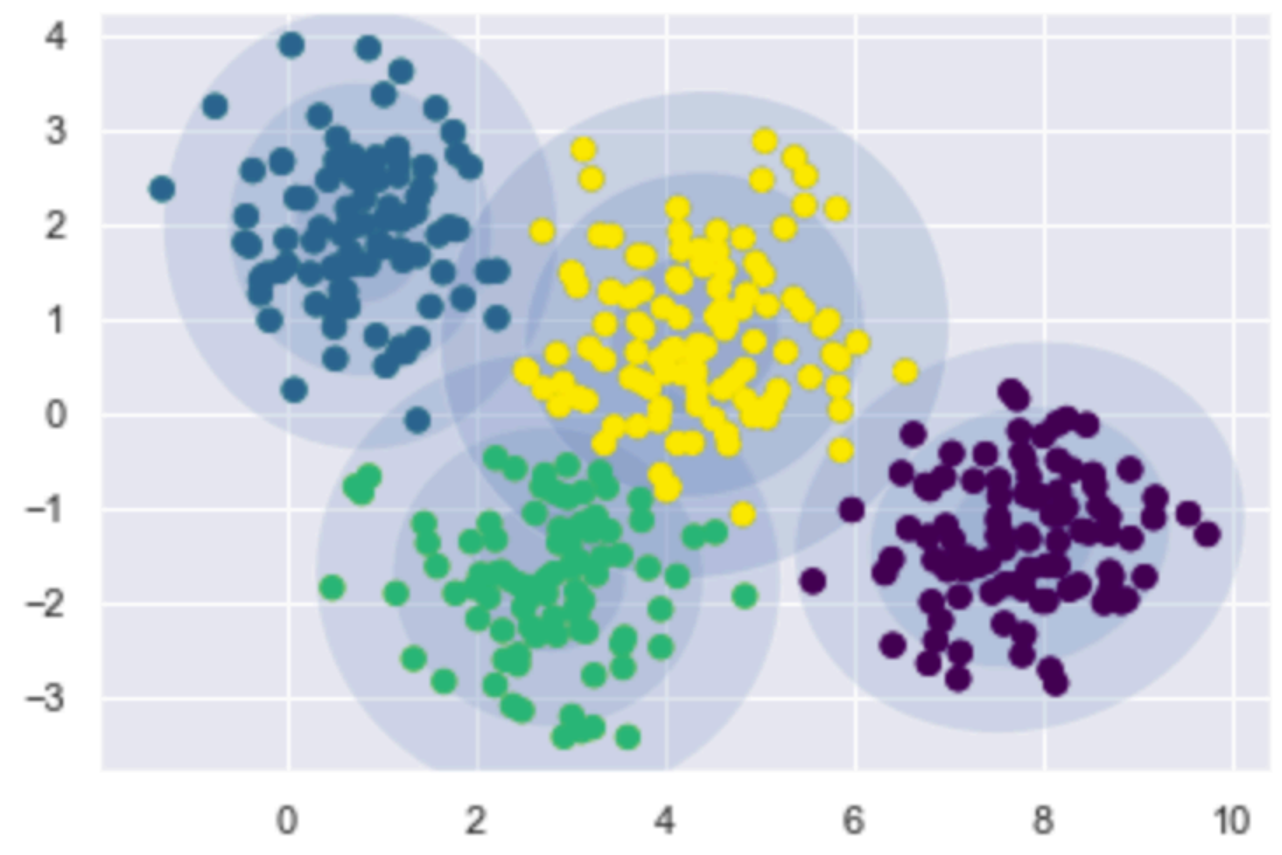


We are now in the following situation :

- **ESTIMATION:**
  If we **knew the parameters**, we could compute the posteriors
- **MAXIMIZATION:**
  If we **knew the posteriors/ sources**, we could easily compute the parameters

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,

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

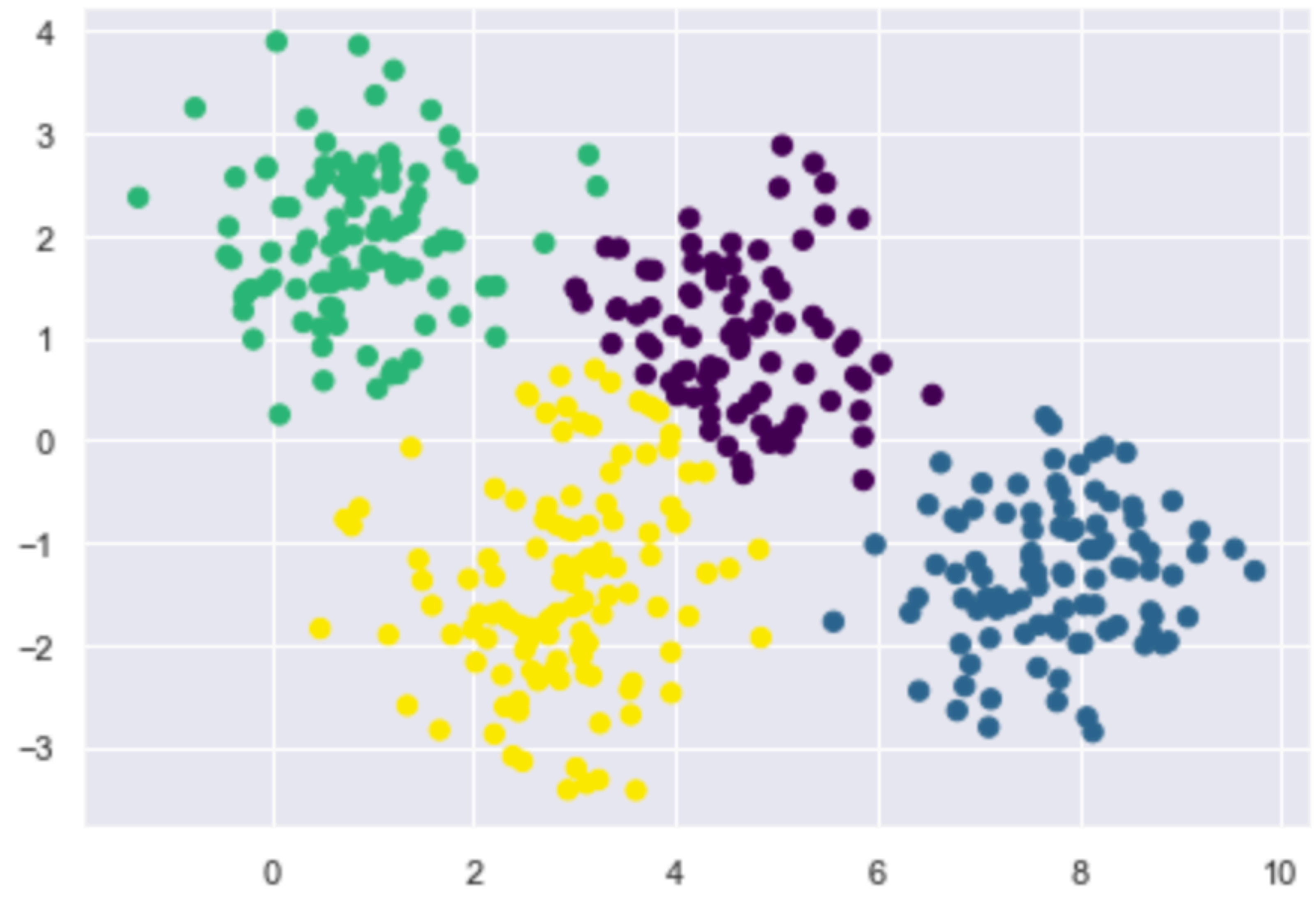
$$\mu^{MLE}_{soft} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

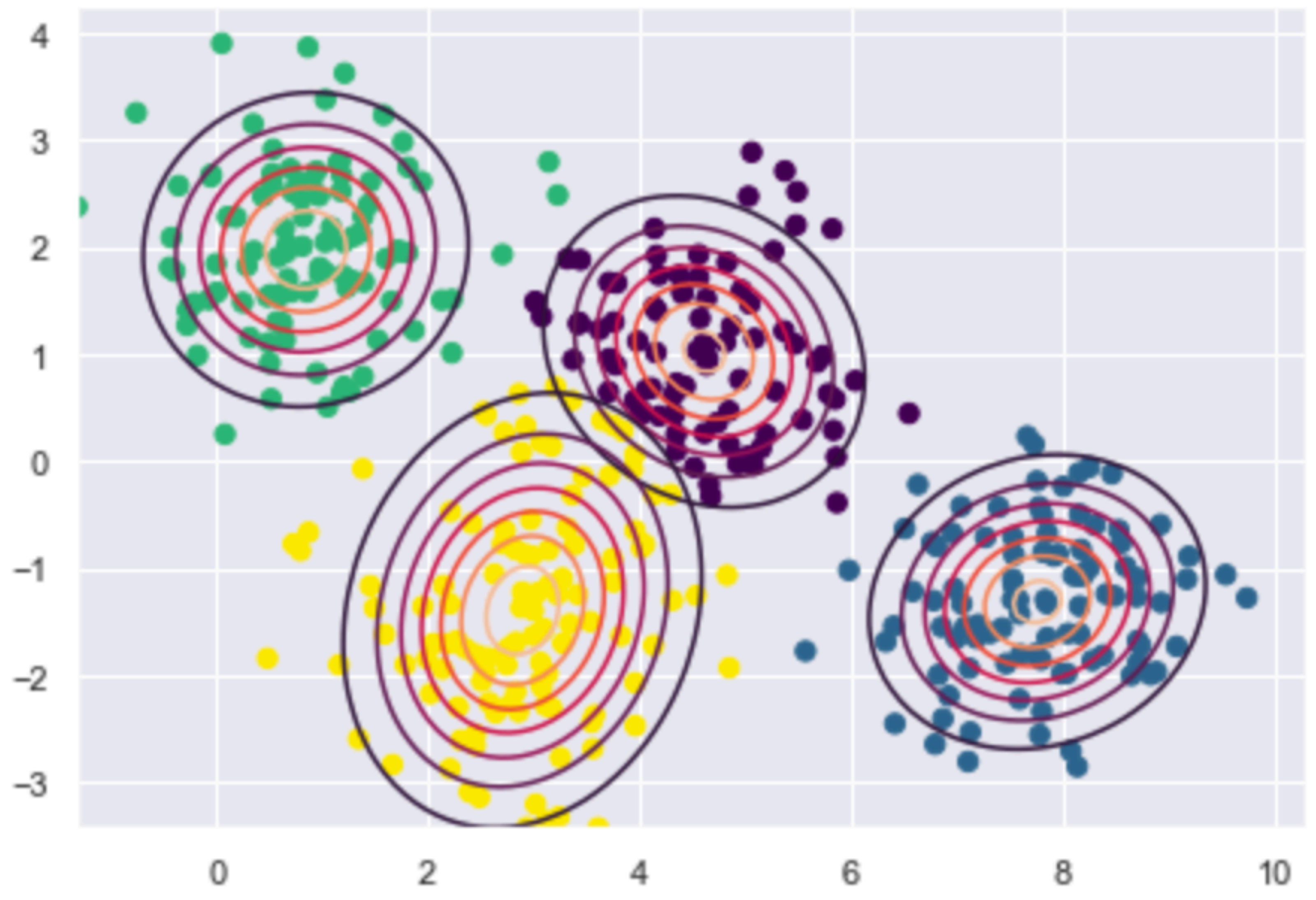
$$\Sigma^{MLE}_{soft} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\,(x_i - \mu^{MLE}_{soft}) \times (x_i - \mu^{MLE}_{soft})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

Remarks: If we **know the parameters** of each instances then**,**

$$p(t = 2 \mid x, \theta) = \frac{p(x \mid t = 2, \theta) \times p(t = 2 \mid \theta)}{\text{Const}}$$

STEP 4

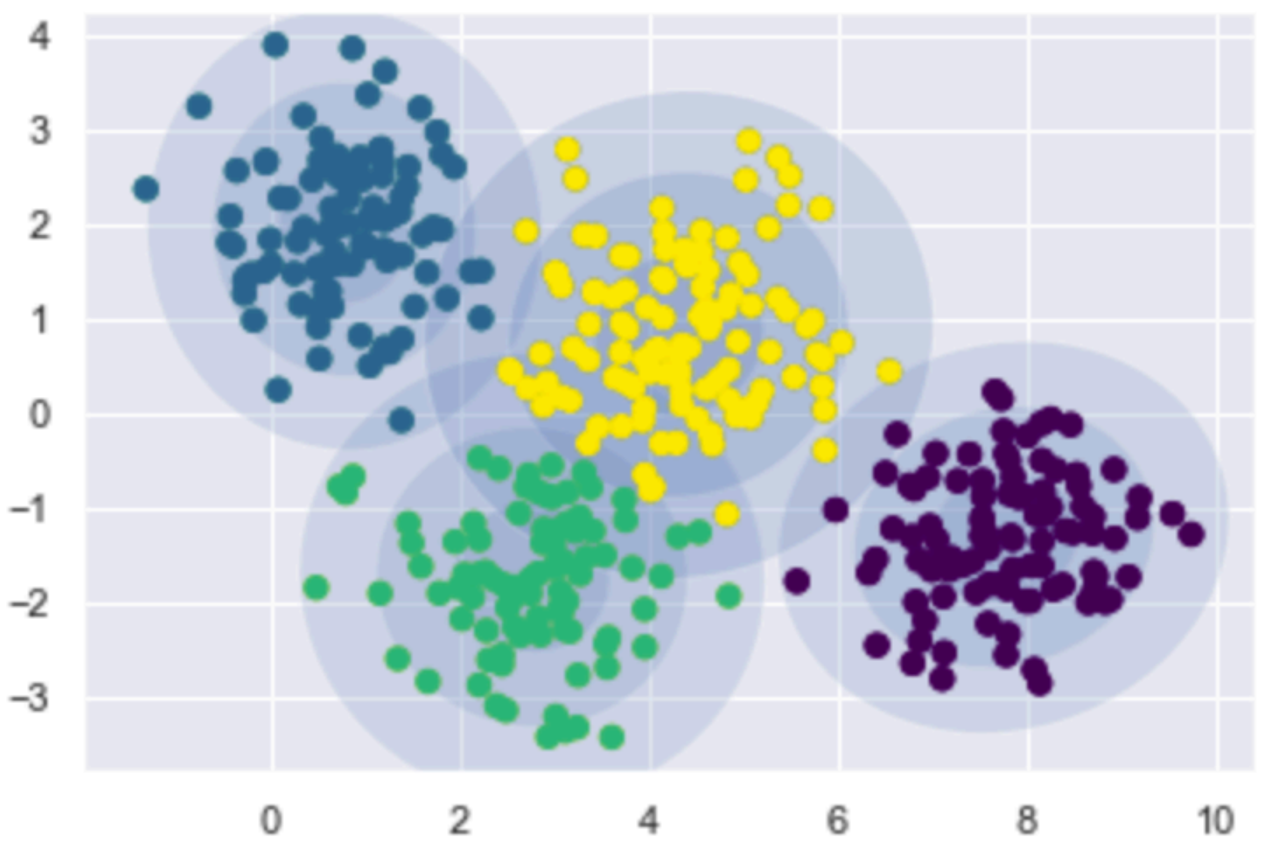




We are now in the following situation :
- **ESTIMATION:**
  If we **knew the parameters**, we could compute the posteriors
- **MAXIMIZATION:**
  If we **knew the posteriors/ sources**, we could easily compute the parameters

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,

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

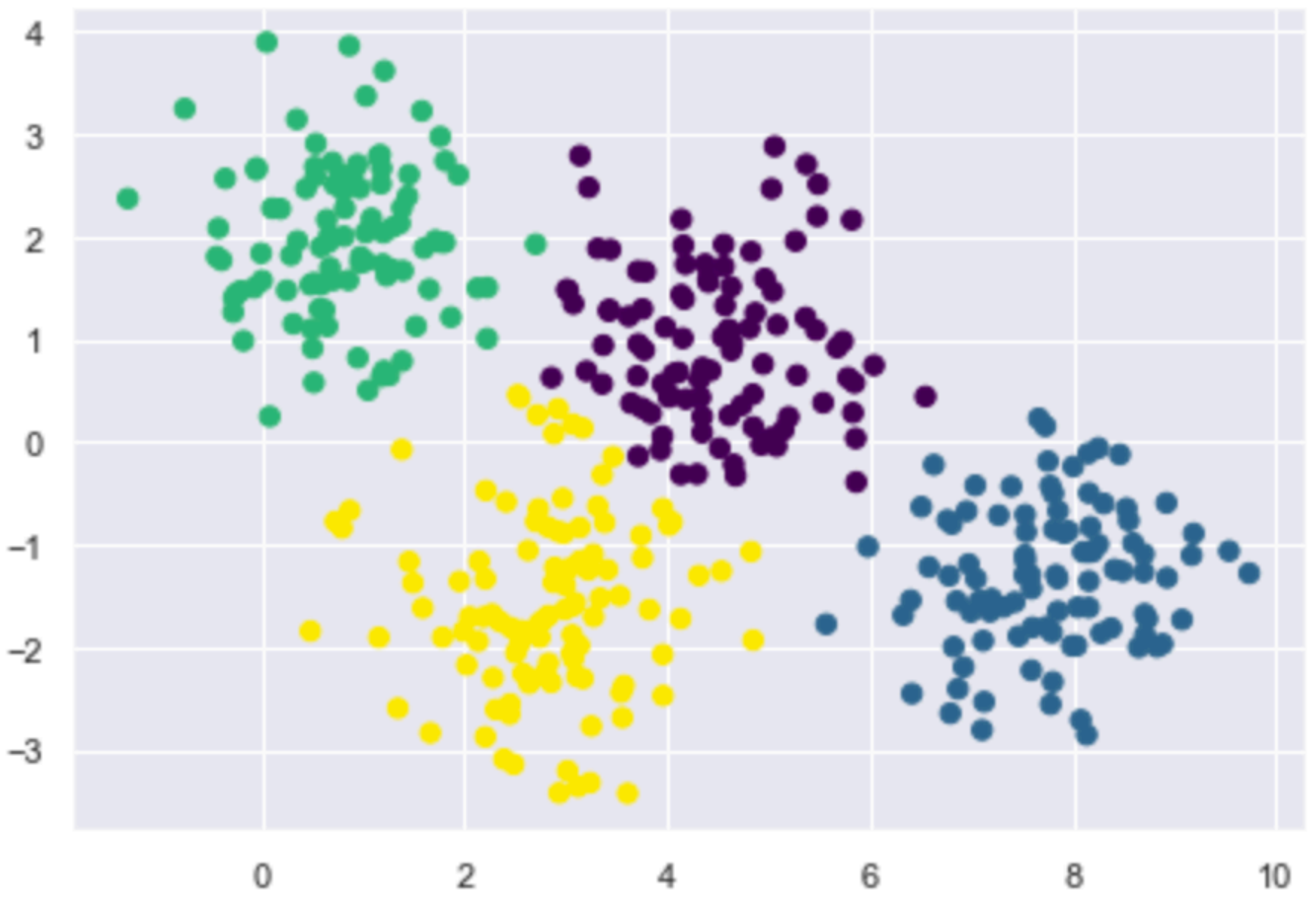
$$\mu^{MLE}_{soft} = \frac{\sum_i \; p(t=2 \,|\, x, \theta)\; x_i}{\sum_i p(\; t=2 \,|\, x, \theta)}$$

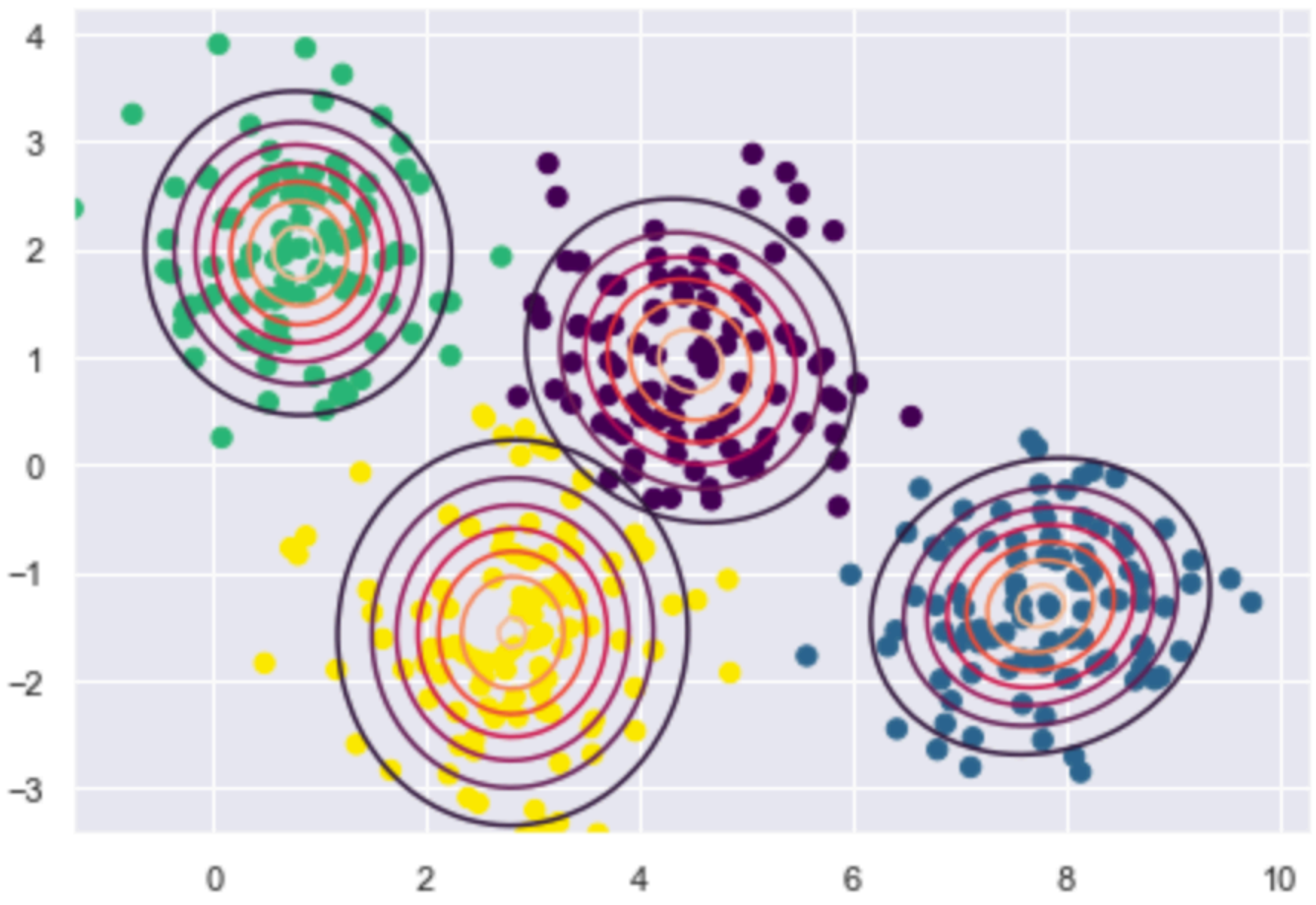
$$\Sigma^{MLE}_{soft} = \frac{\sum_i \; p(t=2 \,|\, x_i, \theta)\,(x_i - \mu^{MLE}_{soft}) \times (x_i - \mu^{MLE}_{soft})^T}{\sum_i p(\; t=2 \,|\, x_i, \theta)}$$

Remarks: If we **know the parameters** of each instances then**,**

$$p(t=2 \,|\, x, \theta) = \frac{p(x \,|\, t=2, \theta) \times p(t=2 \,|\, \theta)}{\text{Const}}$$

STEP 5



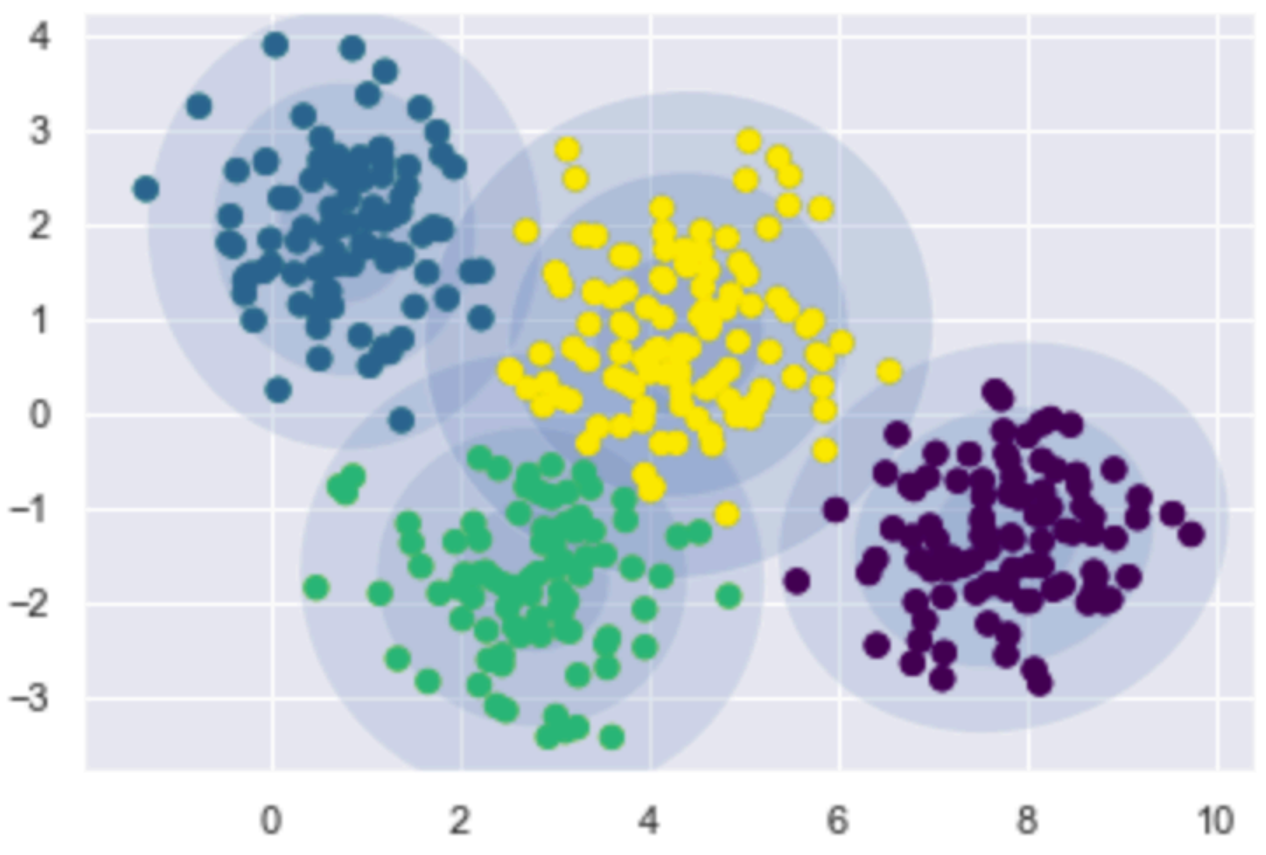


We are now in the following situation :

- **ESTIMATION:**
  If we **knew the parameters**, we could compute the posteriors
- **MAXIMIZATION:**
  If we **knew the posteriors/ sources**, we could easily compute the parameters

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,

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

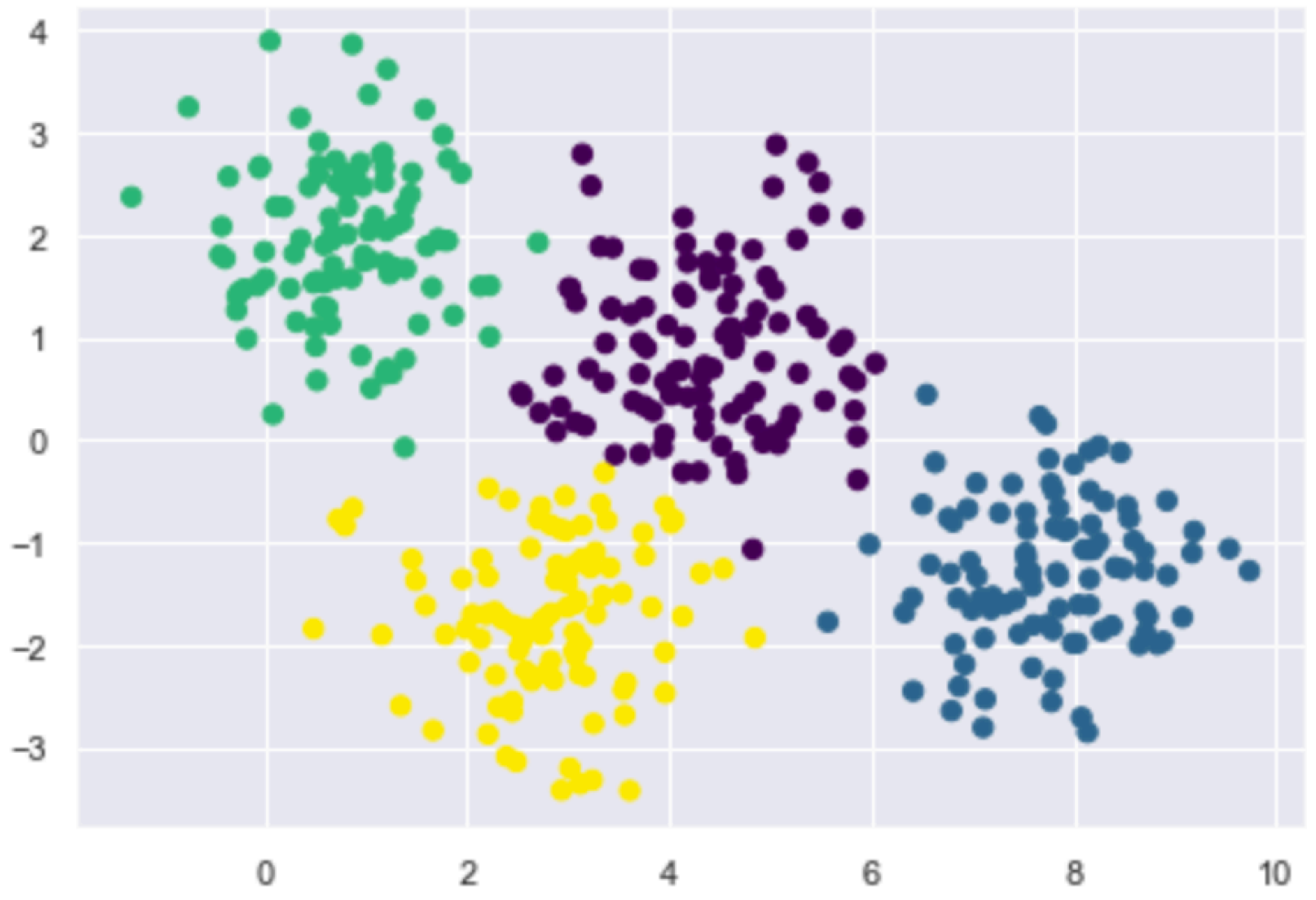
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x, \theta)\, x_i}{\sum_i p(t = 2 \mid x, \theta)}$$

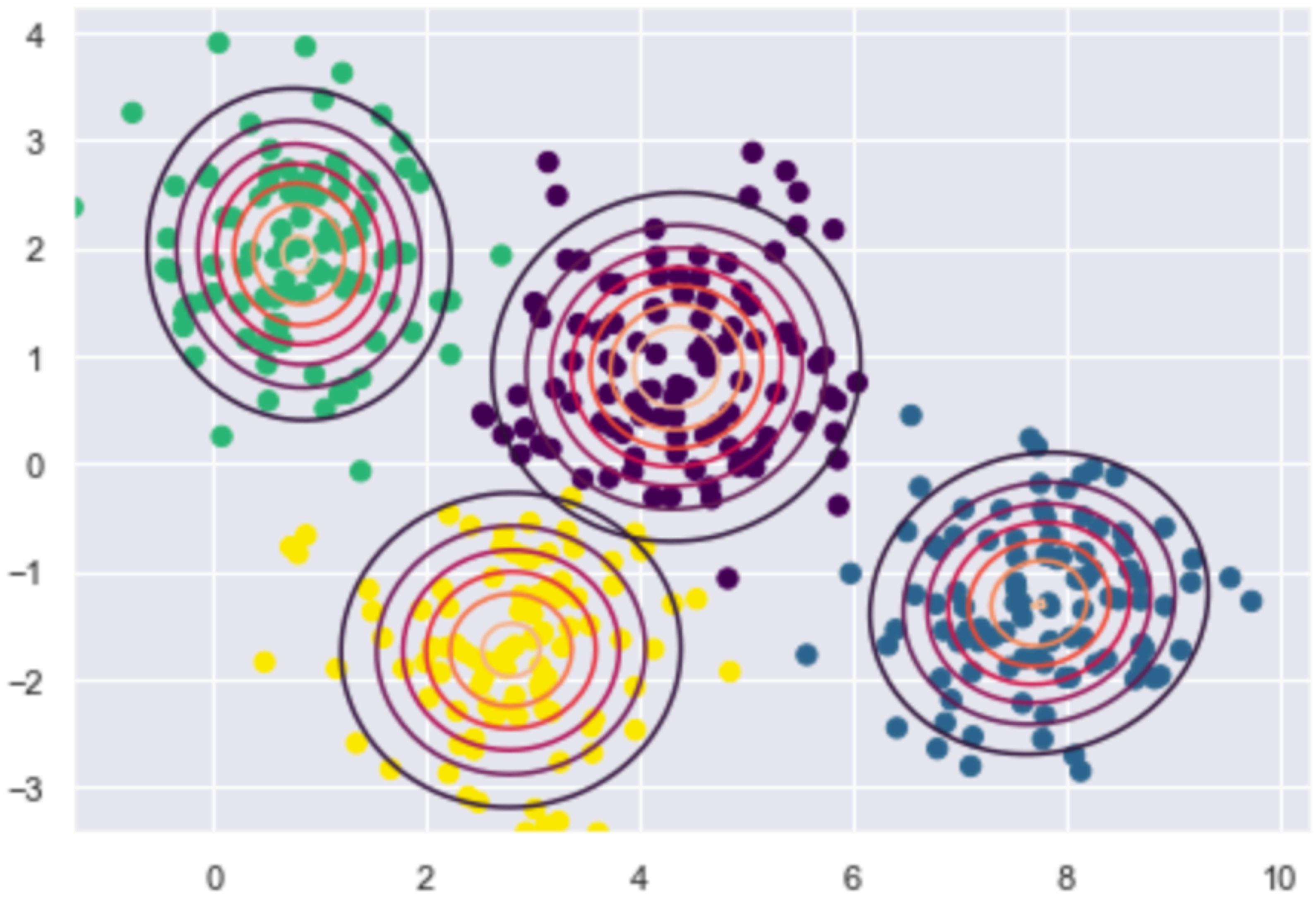
$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t = 2 \mid x_i, \theta)\, (x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t = 2 \mid x_i, \theta)}$$

Remarks: If we **know the parameters** of each instances then**,**

$$p(t = 2 \mid x, \theta) = \frac{p(x \mid t = 2, \theta) \times p(t = 2 \mid \theta)}{\text{Const}}$$

STEP 6



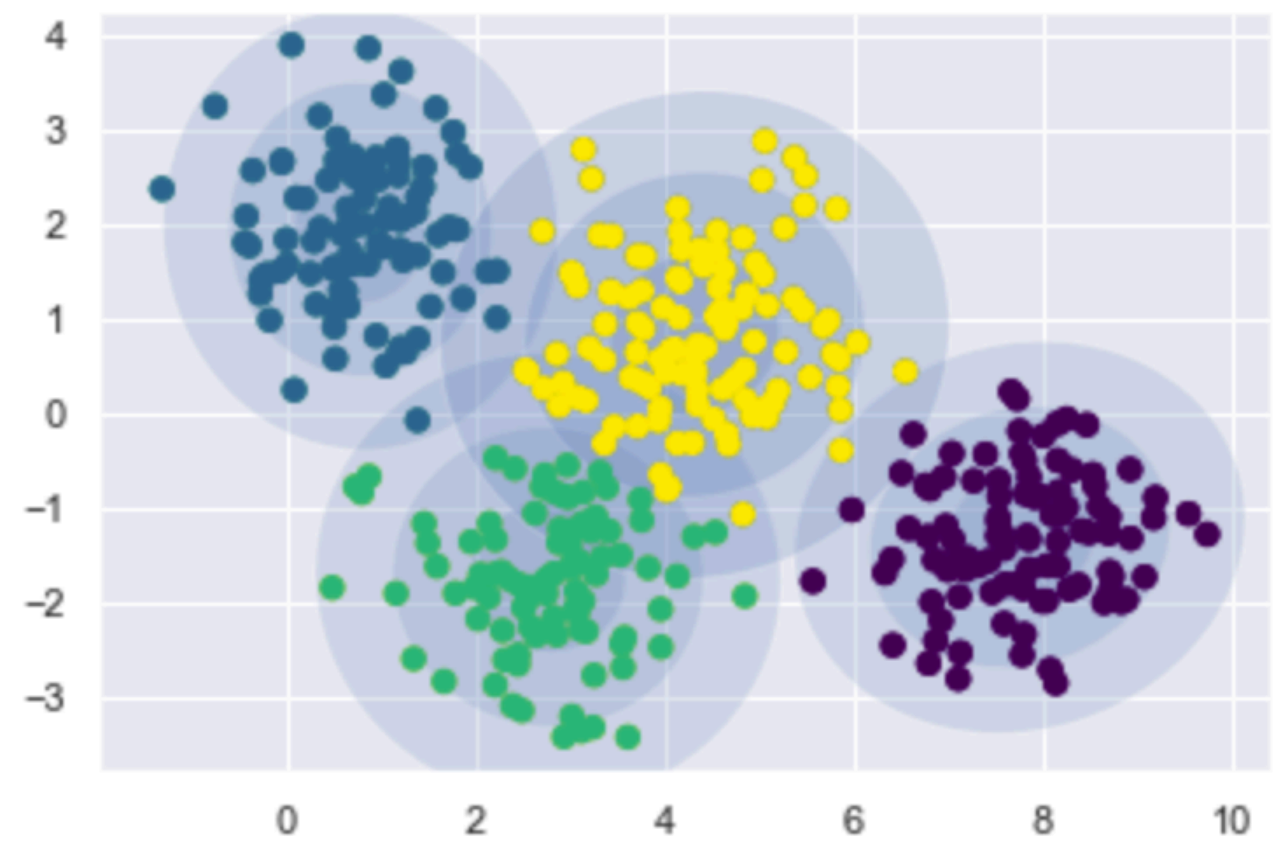


We are now in the following situation :

- **ESTIMATION:**
  If we **knew the parameters**, we could compute the posteriors
- **MAXIMIZATION:**
  If we **knew the posteriors/ sources**, we could easily compute the parameters

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,

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

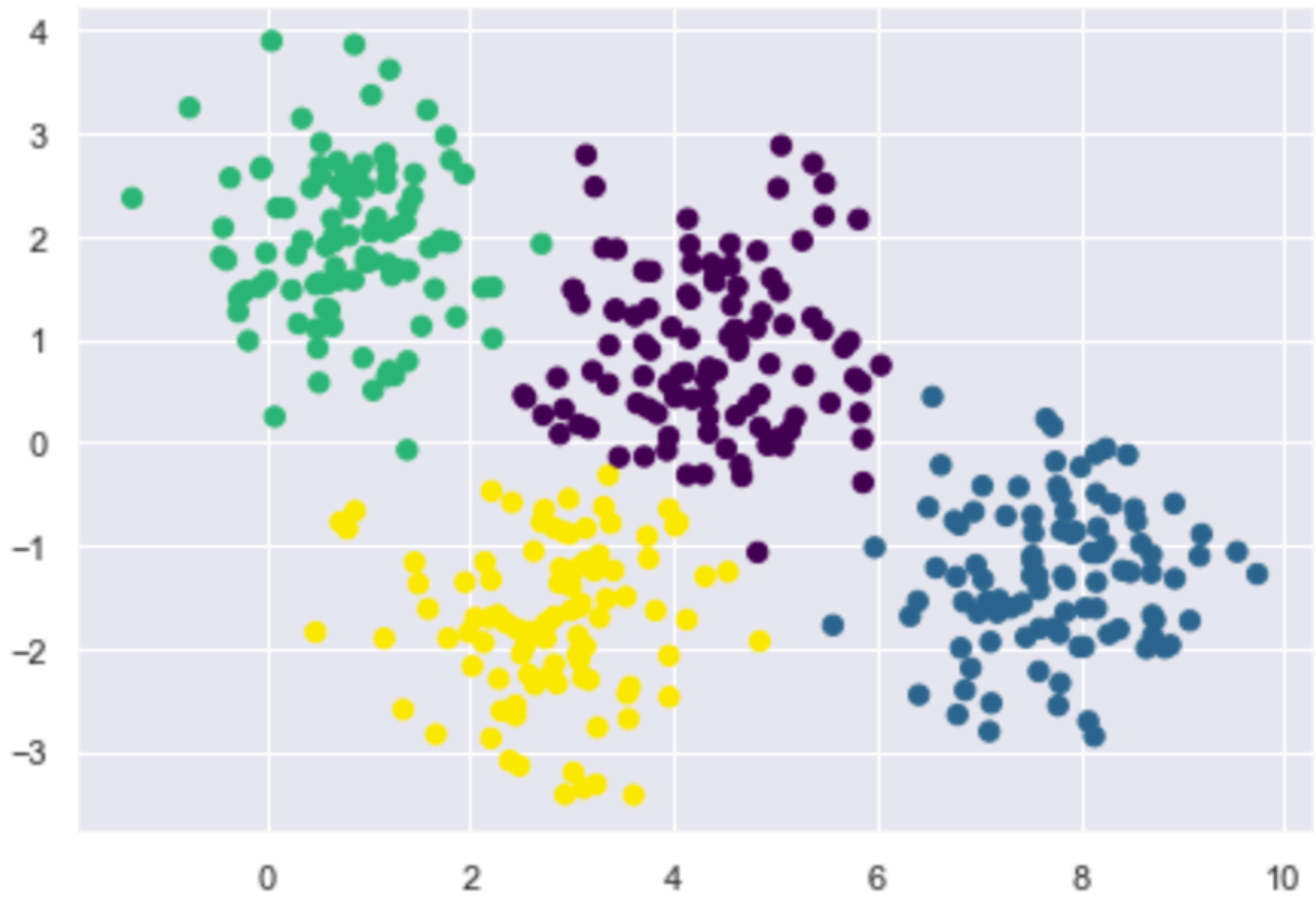
$$\mu_{soft}^{MLE} = \frac{\sum_i p(t=2 \mid x, \theta)\, x_i}{\sum_i p(t=2 \mid x, \theta)}$$

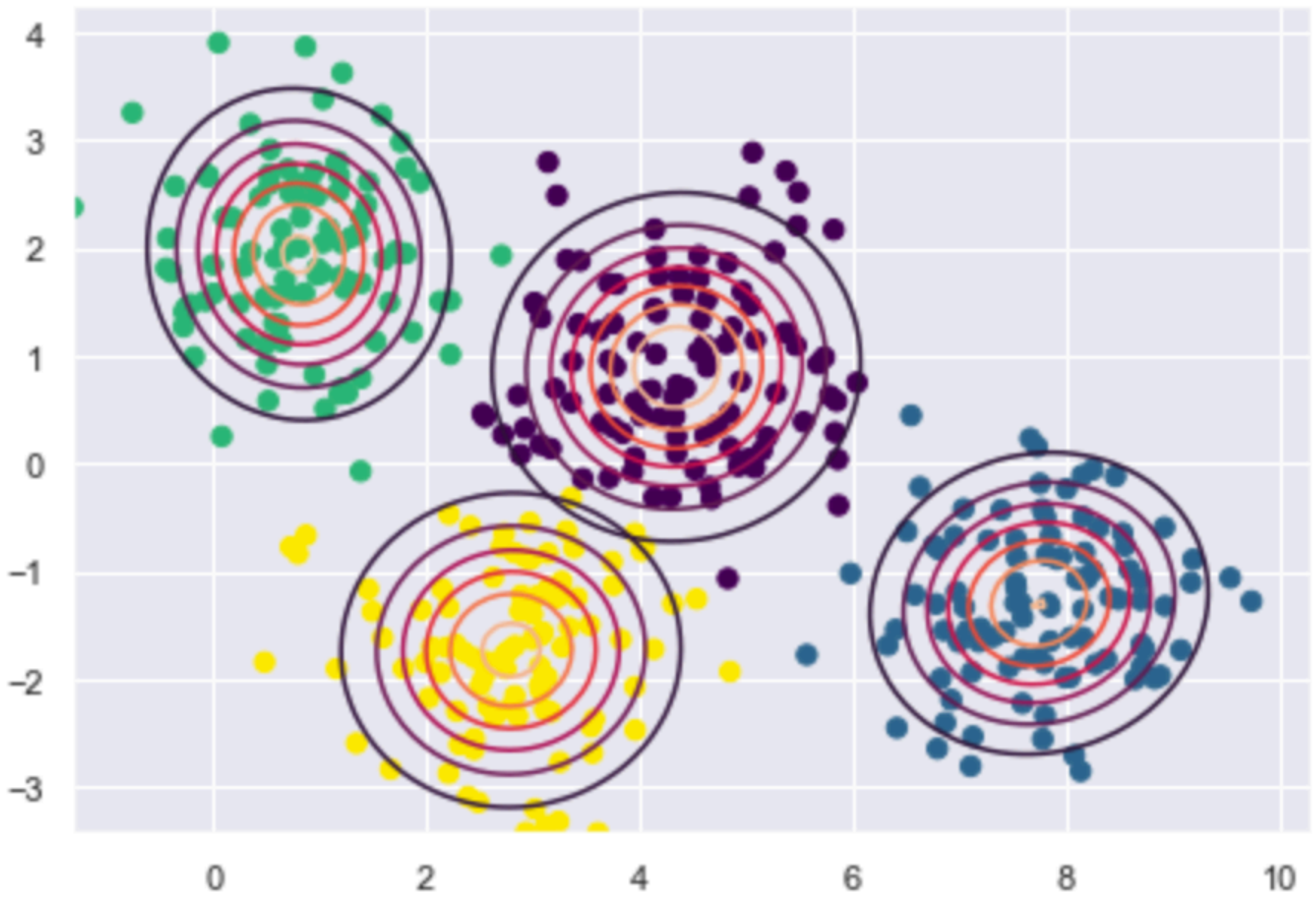
$$\Sigma_{soft}^{MLE} = \frac{\sum_i p(t=2 \mid x_i, \theta)\,(x_i - \mu_{soft}^{MLE}) \times (x_i - \mu_{soft}^{MLE})^T}{\sum_i p(t=2 \mid x_i, \theta)}$$

Remarks: If we **know the parameters** of each instances then**,**

$$p(t=2 \mid x, \theta) = \frac{p(x \mid t=2, \theta) \times p(t=2 \mid \theta)}{\text{Const}}$$

STEP 6





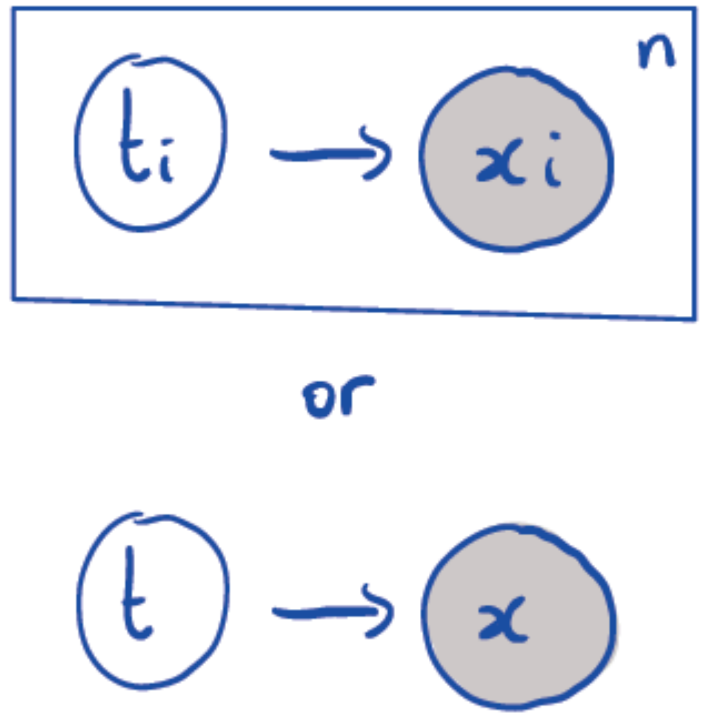
flexible probabilistic approach to clustering problem

2.b EM-algorithm

2.b. Expectation-Maximization algorithm

Reminder : Maximum Likelihood Estimation (MLE)

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



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

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

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

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

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

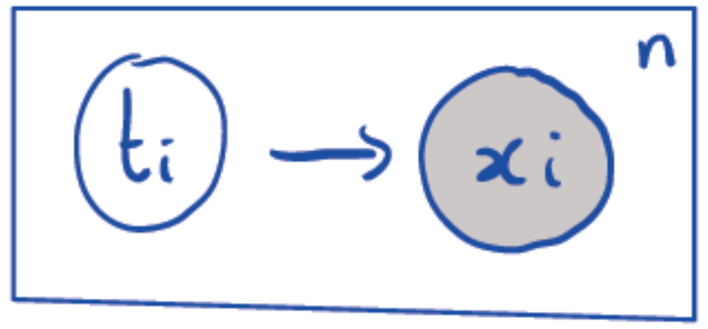
$$\underset{\text{Jensen}}{\geq} \sum_{i=1}^{n} \sum_{k=1}^{4} q(t_i = k) \log \frac{p(x_i, t_i = k \mid \theta)}{q(t_i = k)} \quad \text{for any } q$$

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

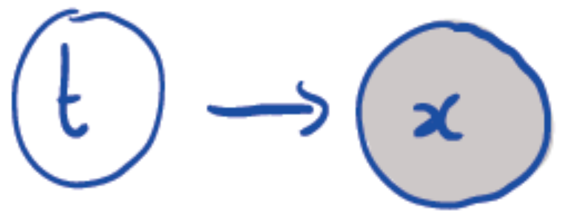
2.b. Expectation-Maximization algorithm

variational lower bound

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



or



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

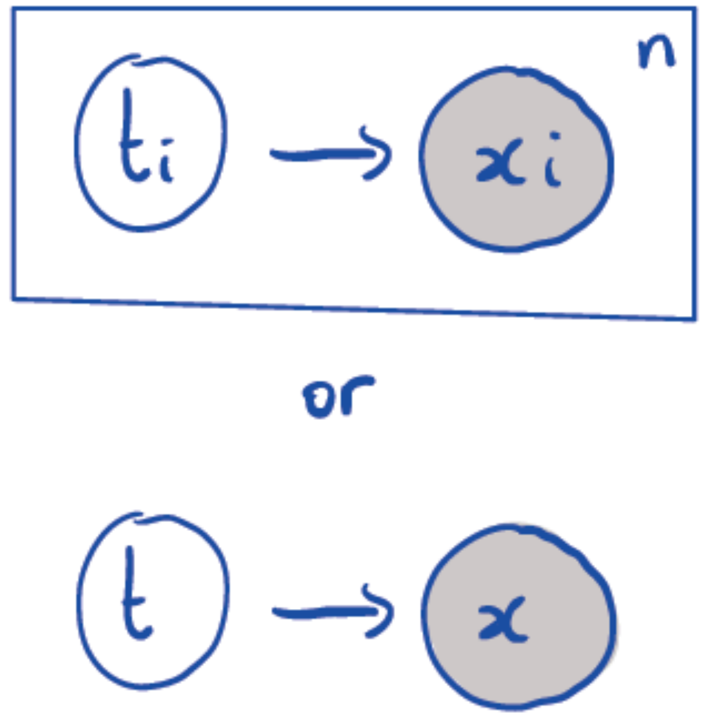
$$\hat{\theta} = \arg\max_{\theta} \left\{ \log P(X \mid \theta) \right\}$$

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

2.b. Expectation-Maximization algorithm

variational lower bound

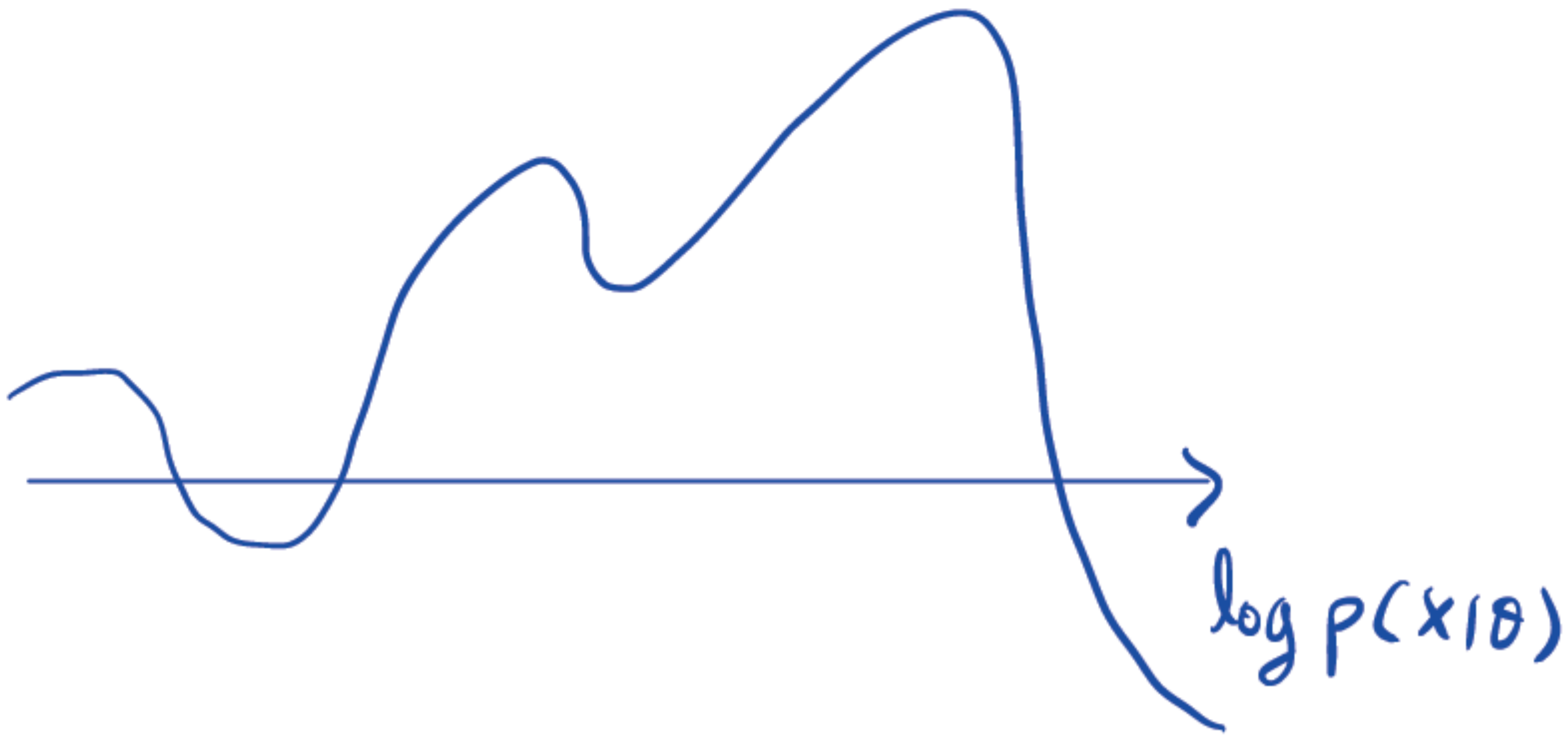
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \mid \theta) = \arg\max_{\theta} \log p(\mathbf{x} \mid \theta)$



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

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

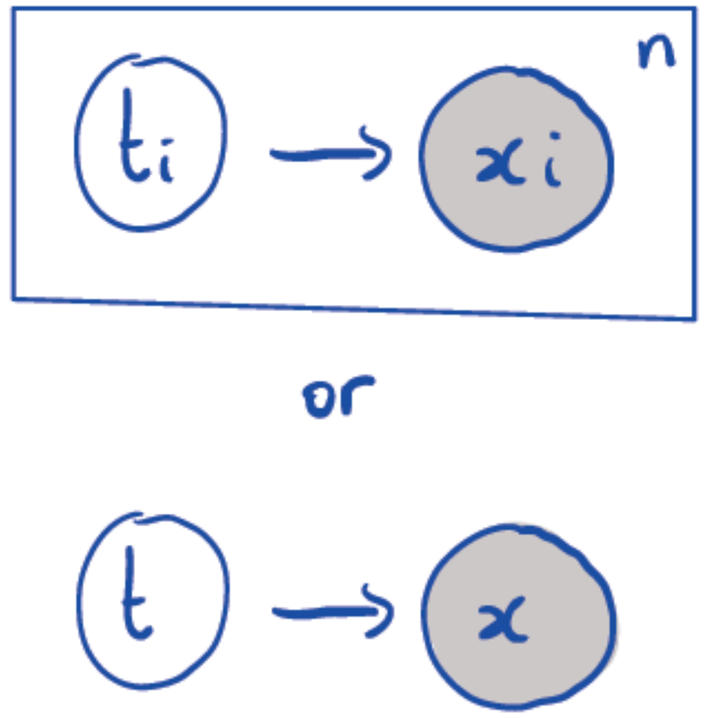
$$\hat{\theta} = \arg\max_{\theta} \left\{ \log P(X \mid \theta) \right\}$$



2.b. Expectation-Maximization algorithm

EM algorithm : E-step

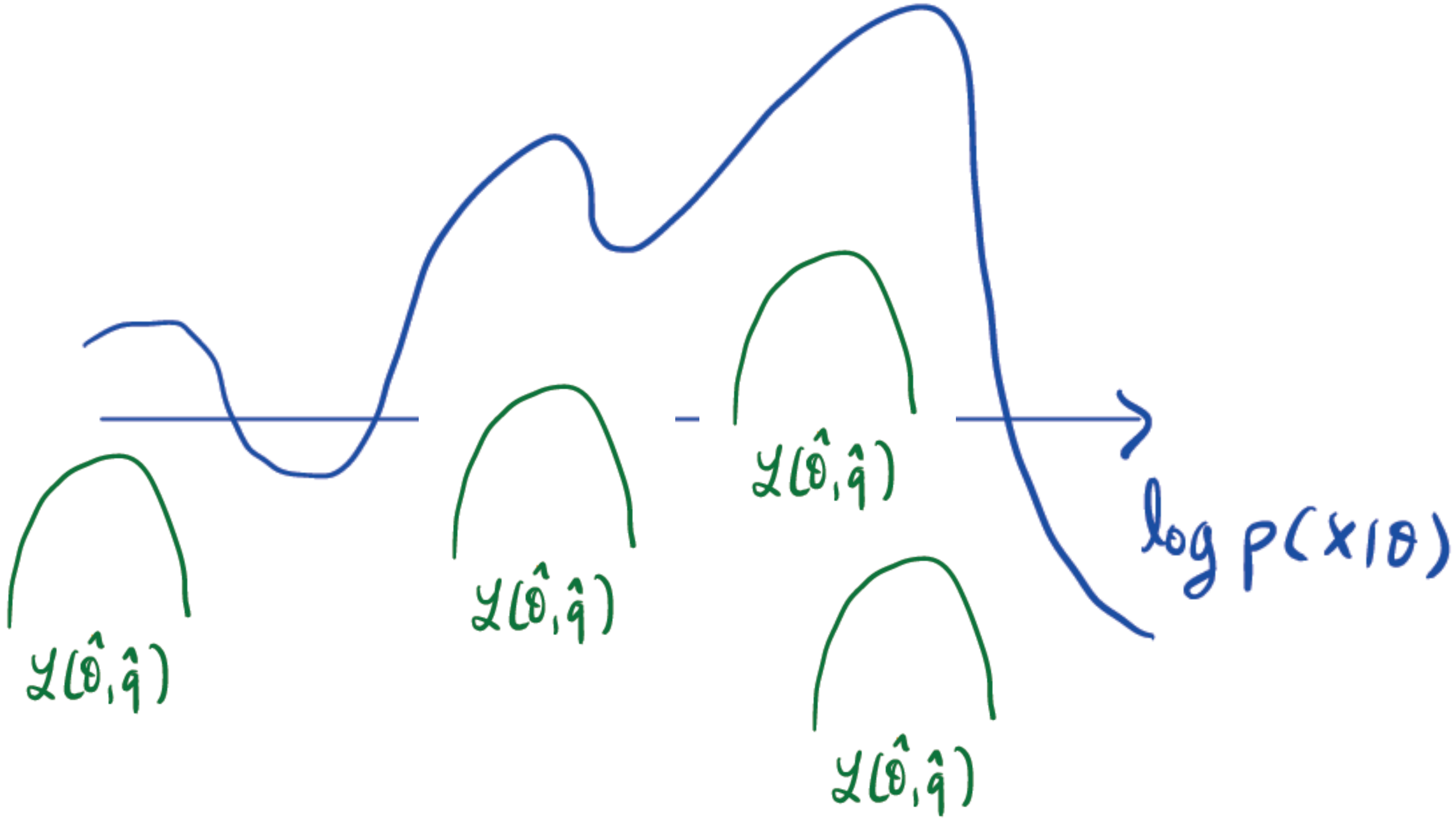
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \mid \theta) = \arg\max_{\theta} \log p(\mathbf{x} \mid \theta)$



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

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

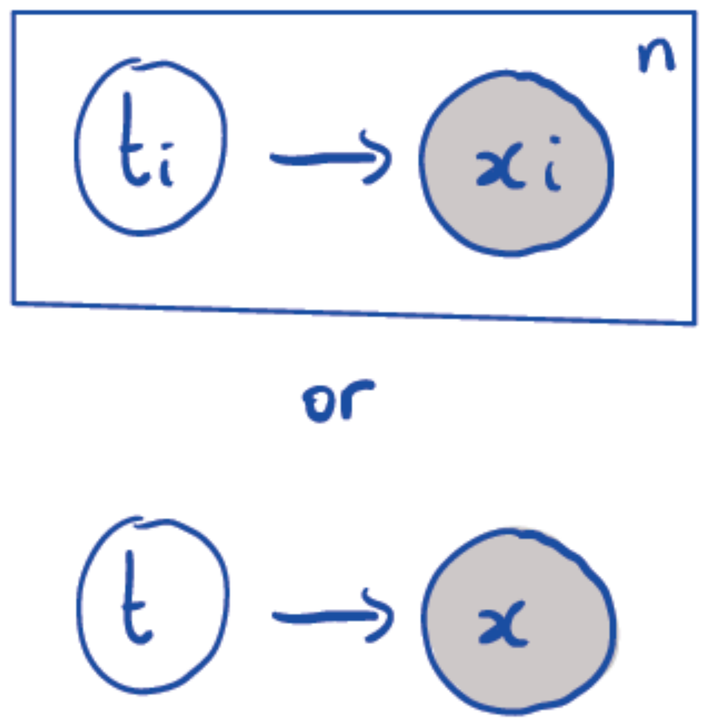
$$\log P(X \mid \theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$$



2.b. Expectation-Maximization algorithm

EM algorithm : E-step

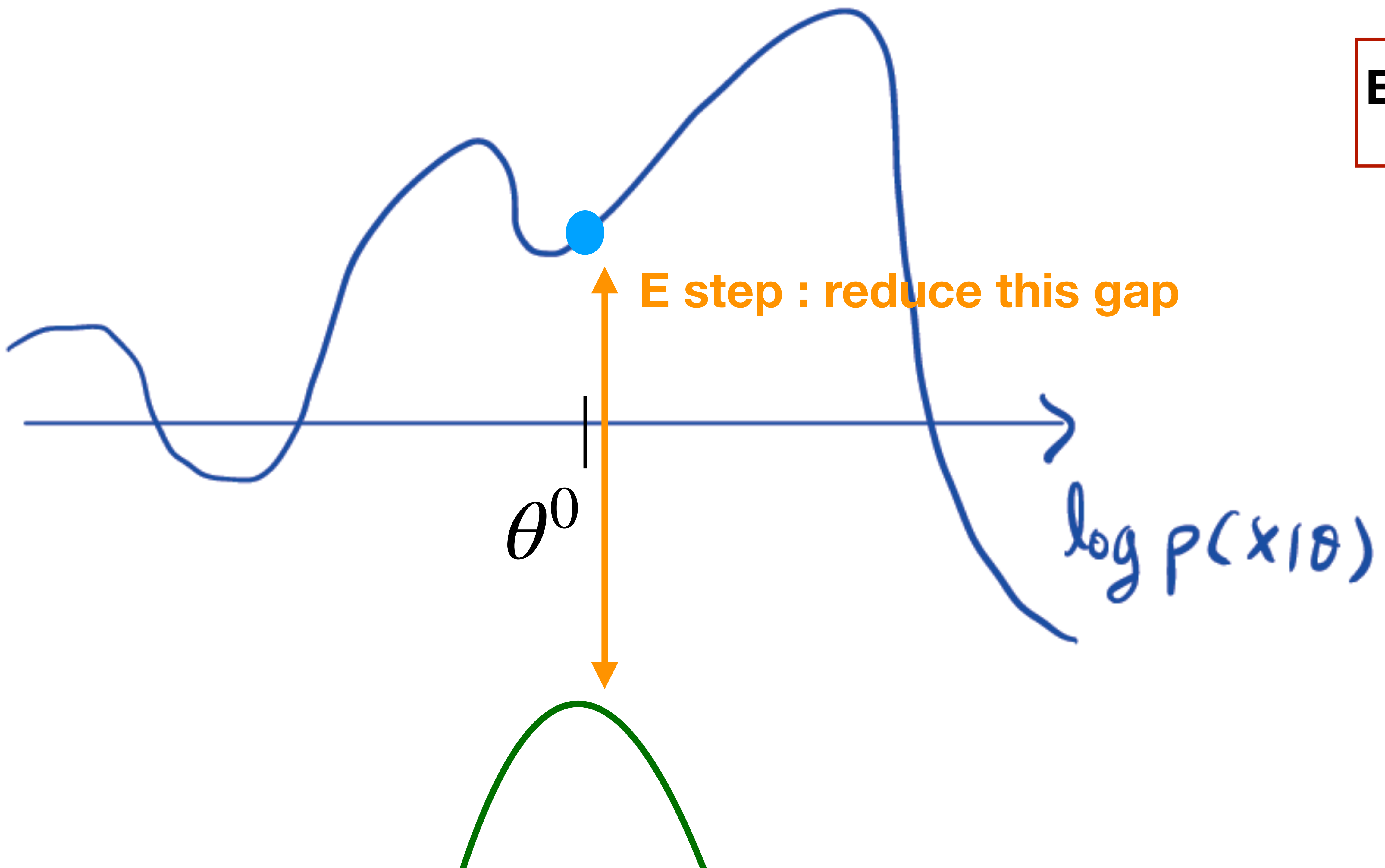
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \mid \theta) = \arg\max_{\theta} \log p(\mathbf{x} \mid \theta)$



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

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

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



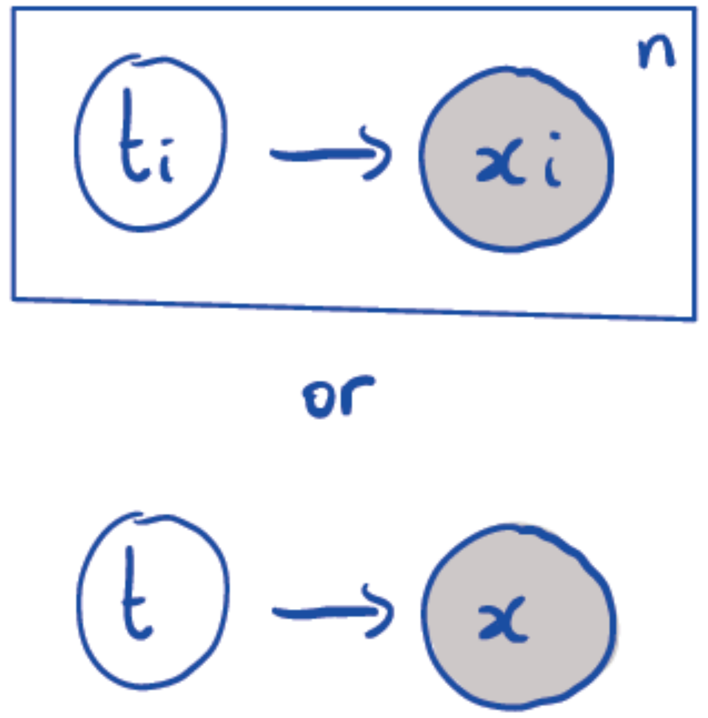
$$\boxed{\textbf{Expectation step : } q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)}$$

2.b. Expectation-Maximization algorithm

EM algorithm : E-step

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

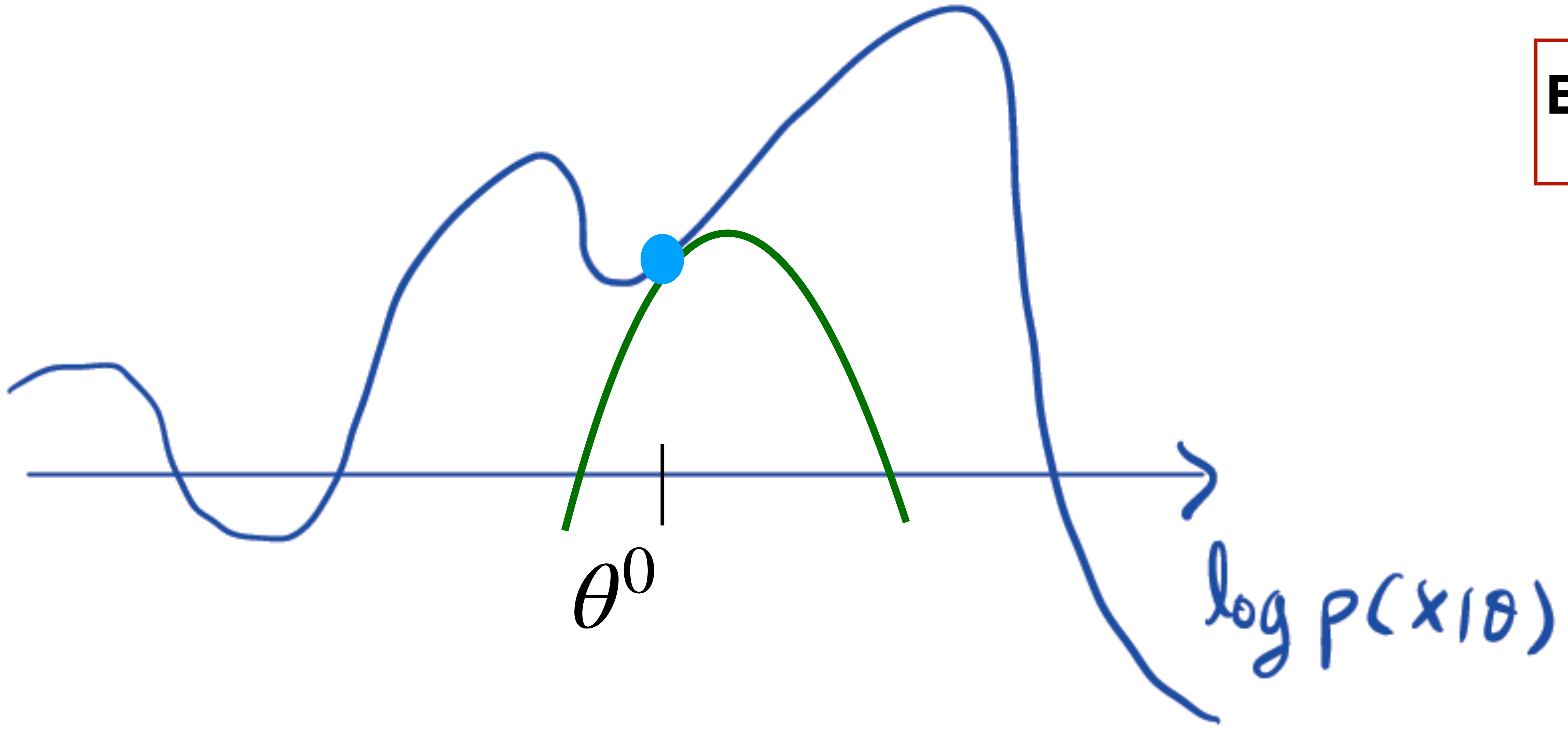
$\log P(x|\theta) \underset{\text{Jensen}}{\geq} \mathcal{L}(\theta, q)$



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

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

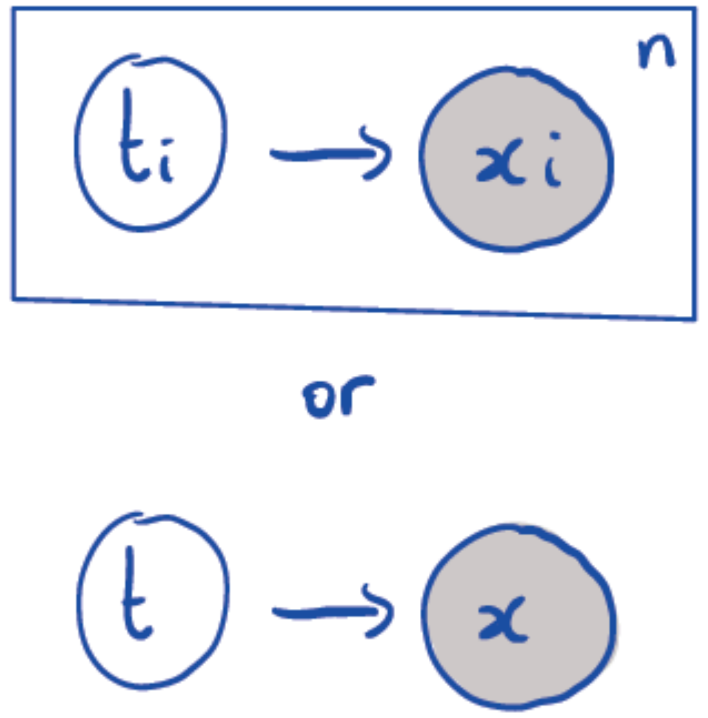
Expectation step : $q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)$



2.b. Expectation-Maximization algorithm

EM algorithm : M-step

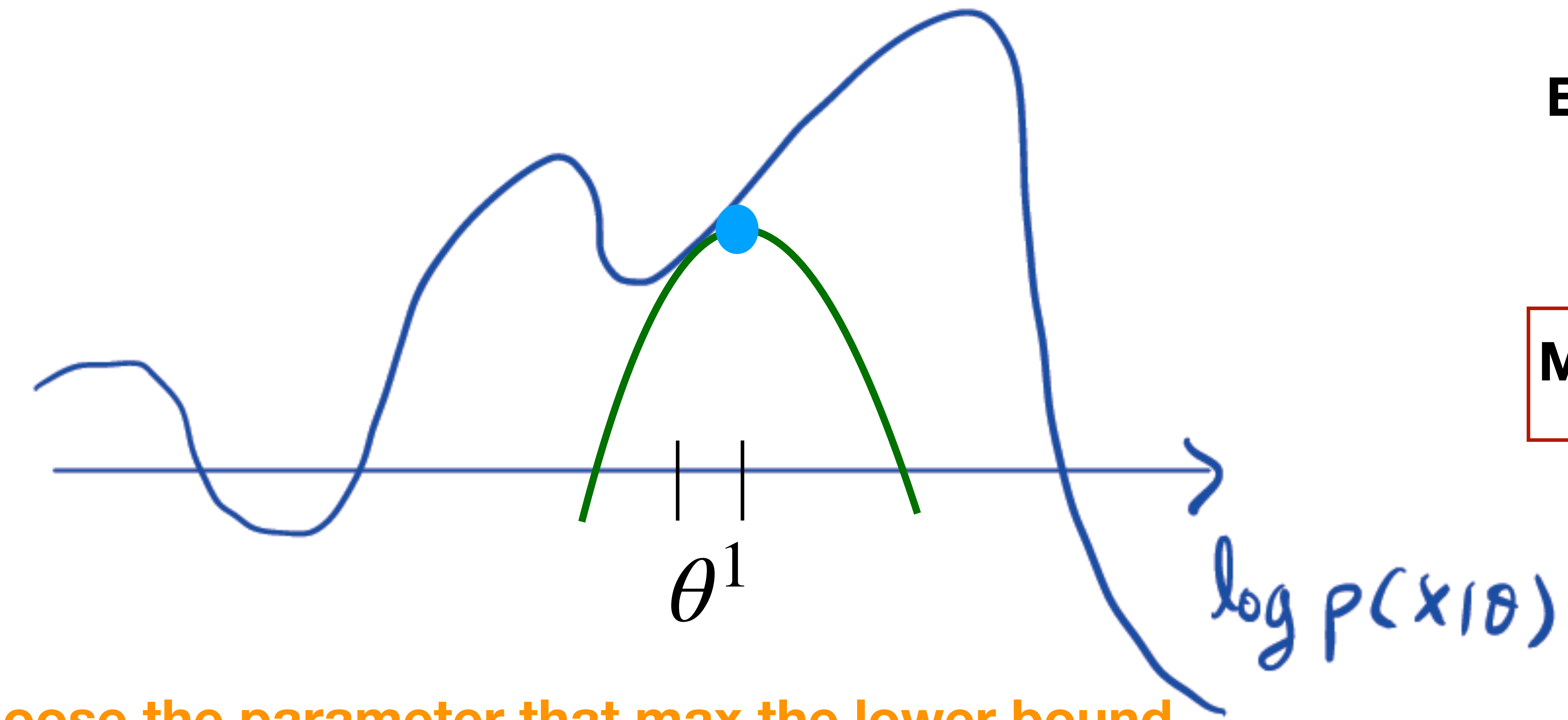
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$



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

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

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



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

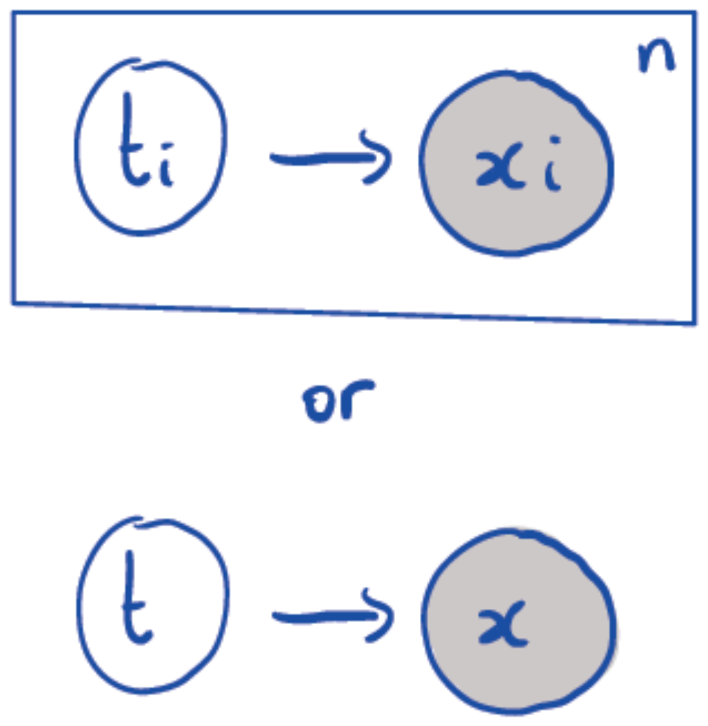
$$\boxed{\textbf{Maximization step : } \theta^{k+1} = \arg\max_{\theta} \mathscr{L}(\theta, q^{k+1})}$$

M step : choose the parameter that max the lower bound

2.b. Expectation-Maximization algorithm

EM algorithm

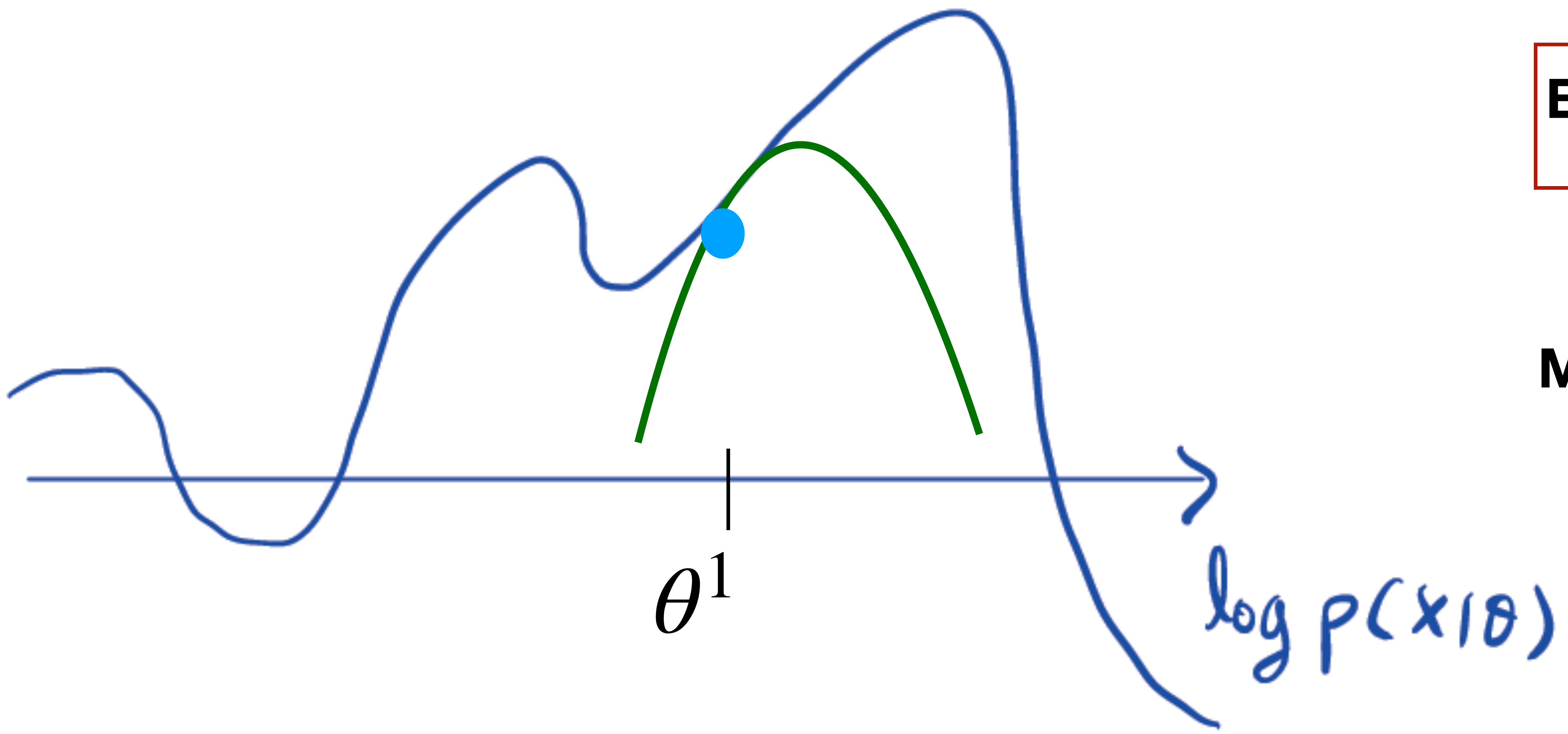
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \,|\, \theta) = \arg\max_{\theta} \log p(\mathbf{x} \,|\, \theta)$



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

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

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



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

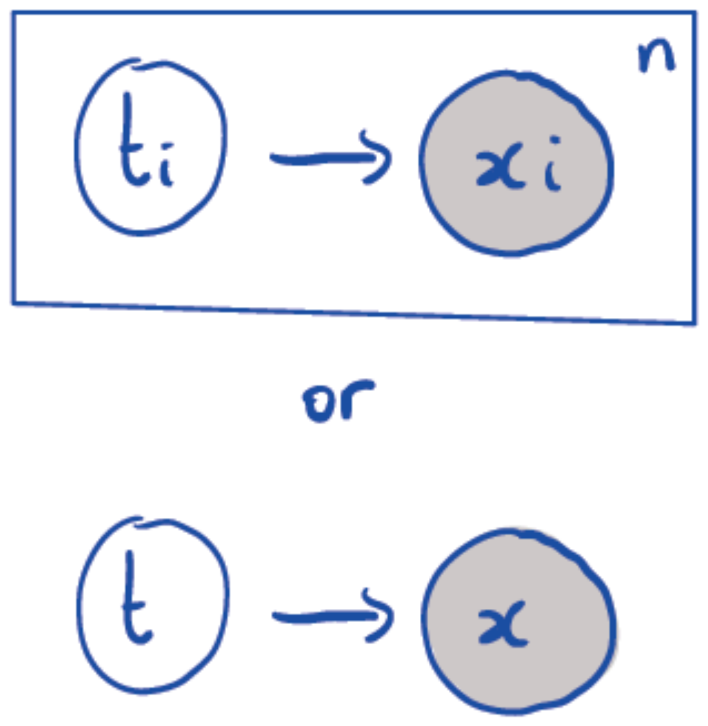
Maximization step : $\theta^{k+1} = \arg\max_{\theta} \mathscr{L}(\theta, q^{k+1})$

2.b. Expectation-Maximization algorithm

EM algorithm

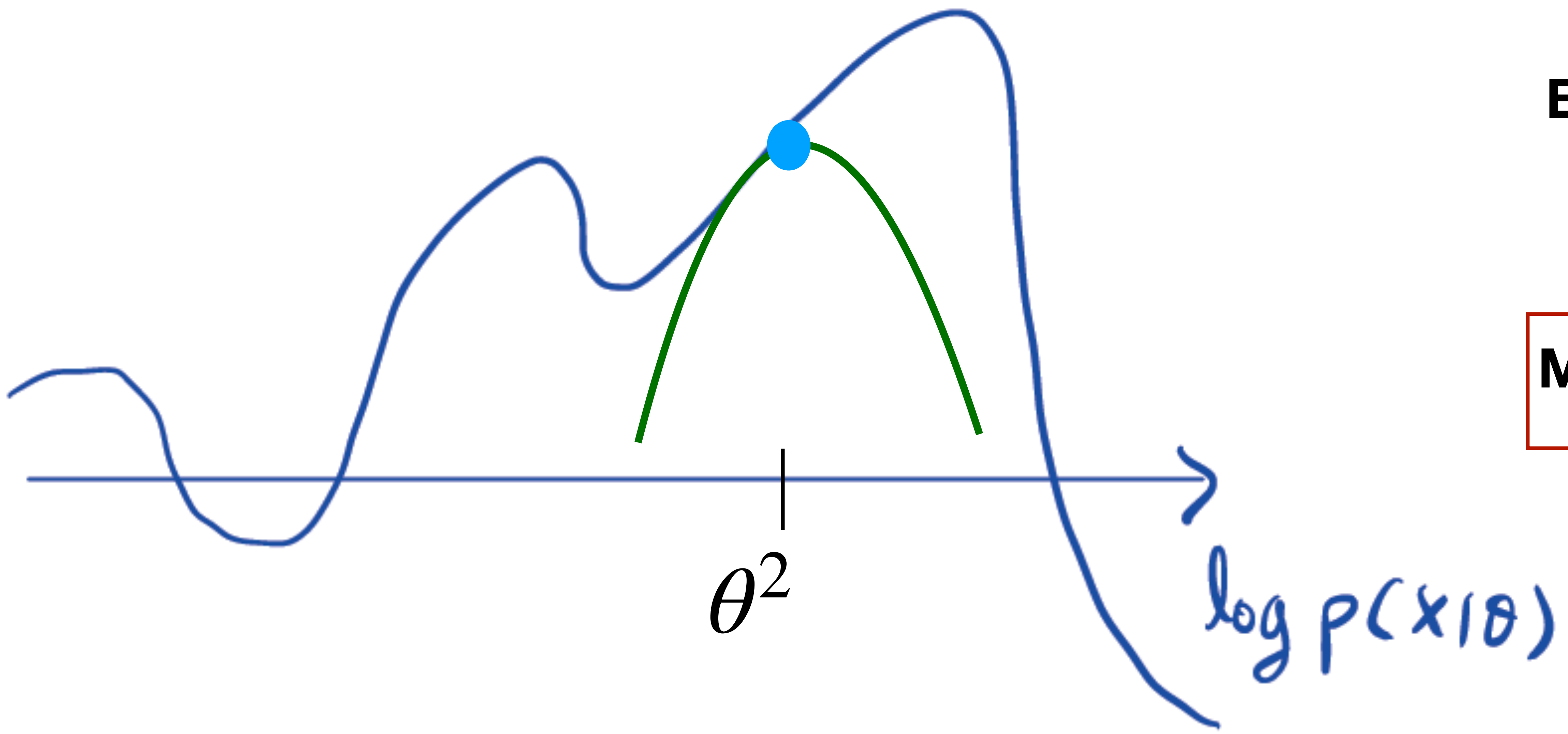
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$

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



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

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



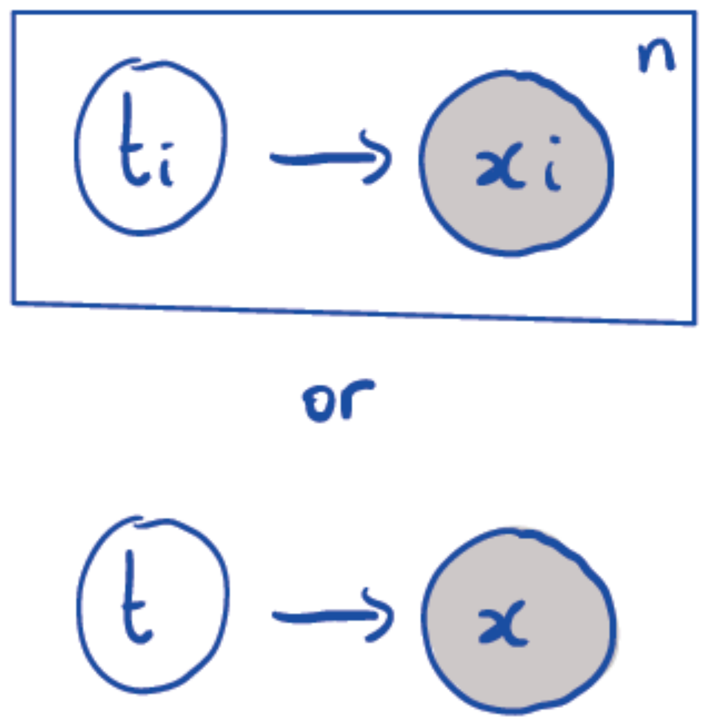
Expectation step : $q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)$

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

2.b. Expectation-Maximization algorithm

EM algorithm

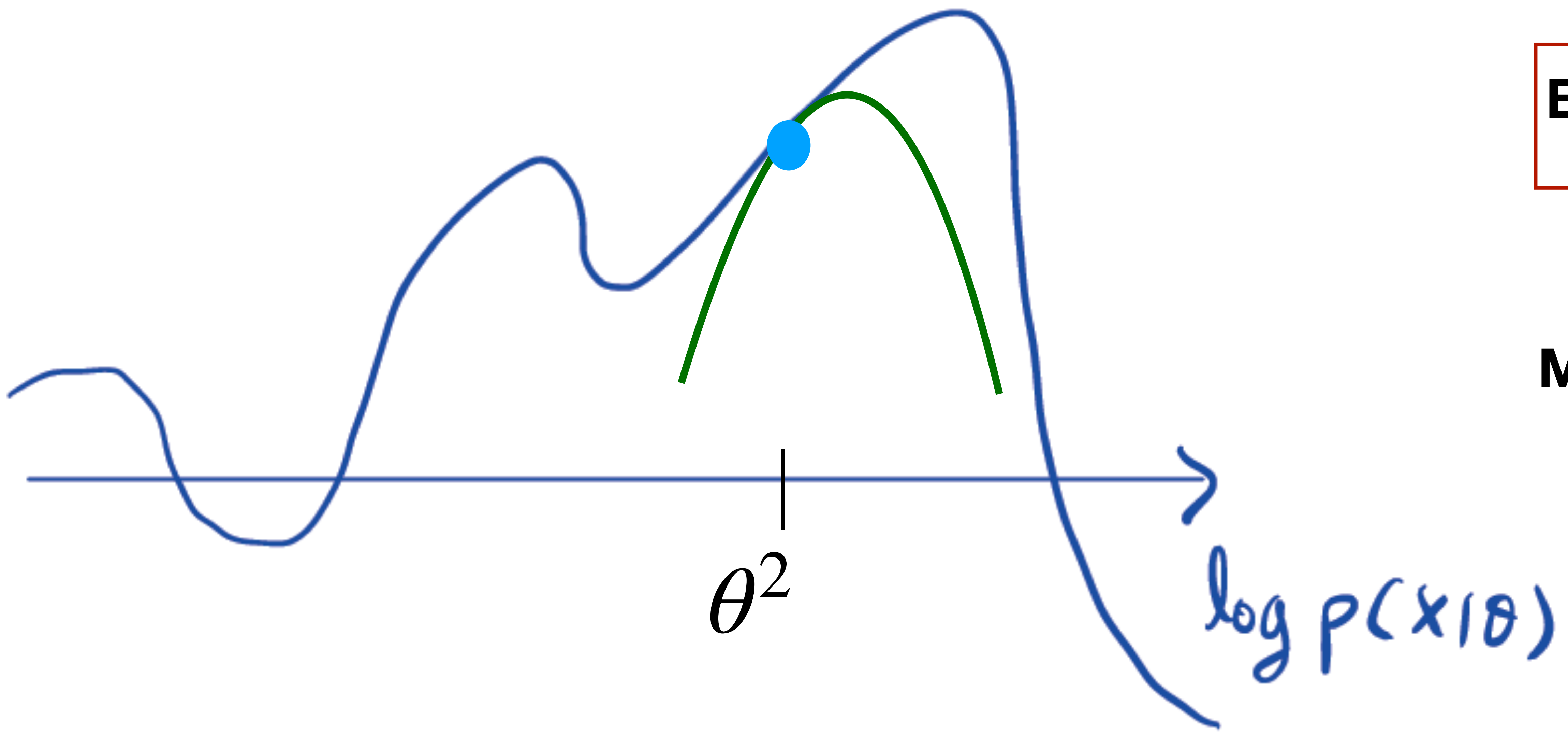
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x} \mid \theta) = \arg\max_{\theta} \log p(\mathbf{x} \mid \theta)$



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

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

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



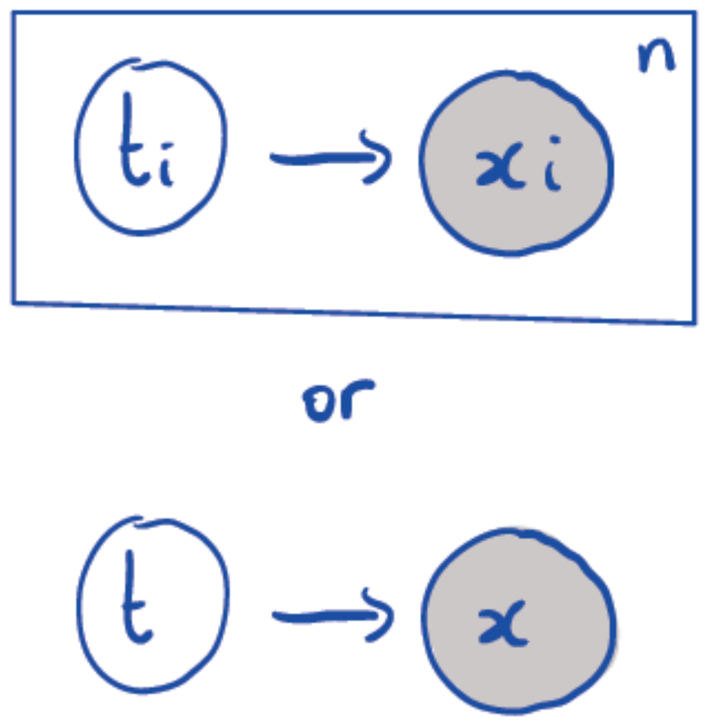
$\boxed{\textbf{Expectation step : } q^{k+1} = \arg \max_{q \in Family} \mathscr{L}(\theta^k, q)}$

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

2.b. Expectation-Maximization algorithm

EM algorithm

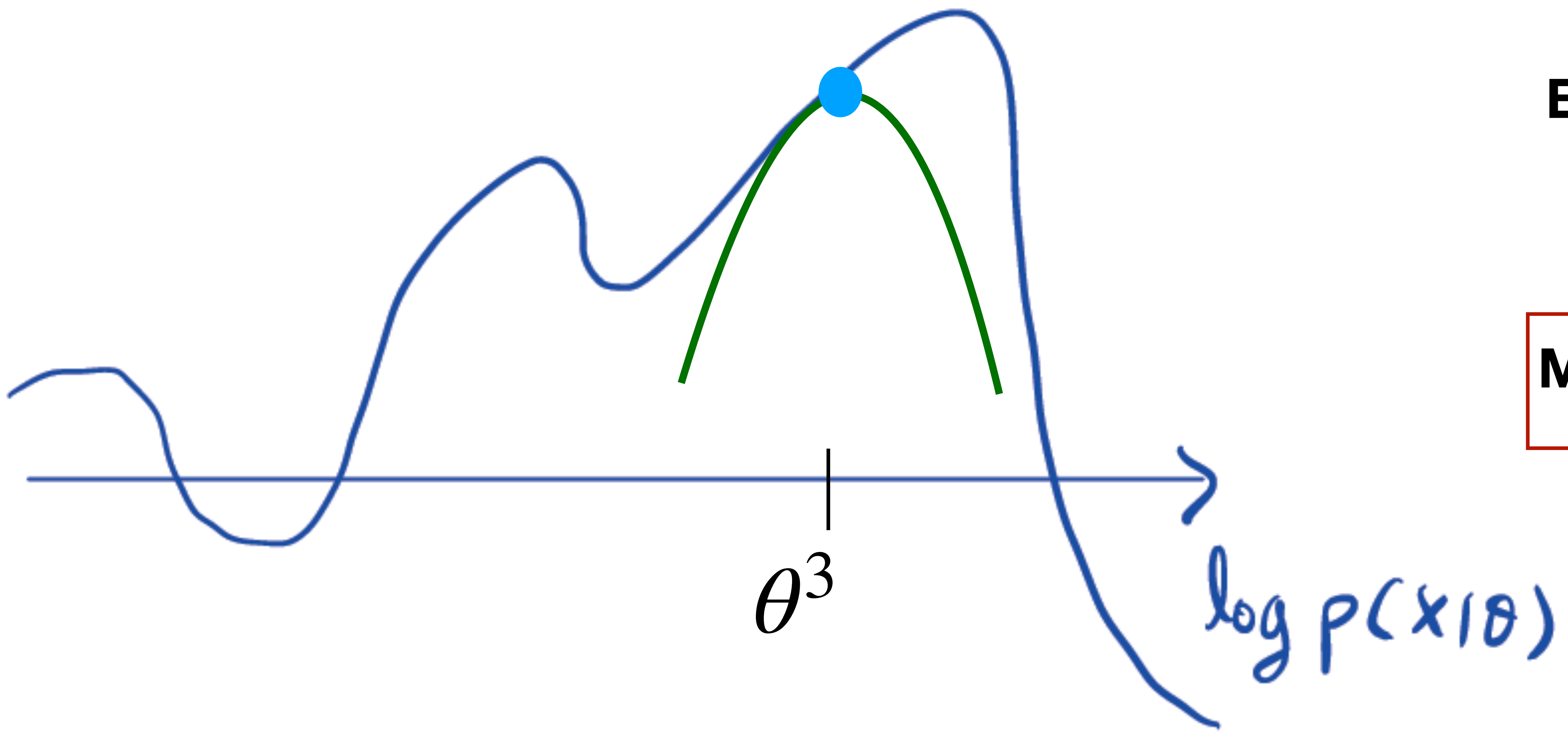
Our aim is to find : $\hat{\theta}^{MLE} = \arg\max_{\theta} p(\mathbf{x}\,|\,\theta) = \arg\max_{\theta} \log p(\mathbf{x}\,|\,\theta)$



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

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

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



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

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

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 :

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

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

M-step :

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

GMM : we updated the gaussian parameters with

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

which indeed is the M-step of the EM algorithm

2.b. Expectation-Maximization algorithm

EM algorithm : back to GMM

E-step :

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

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

M-step :

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

GMM : we updated the gaussian parameters with

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

which indeed is the M-step of the EM algorithm

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

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

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

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

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

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

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

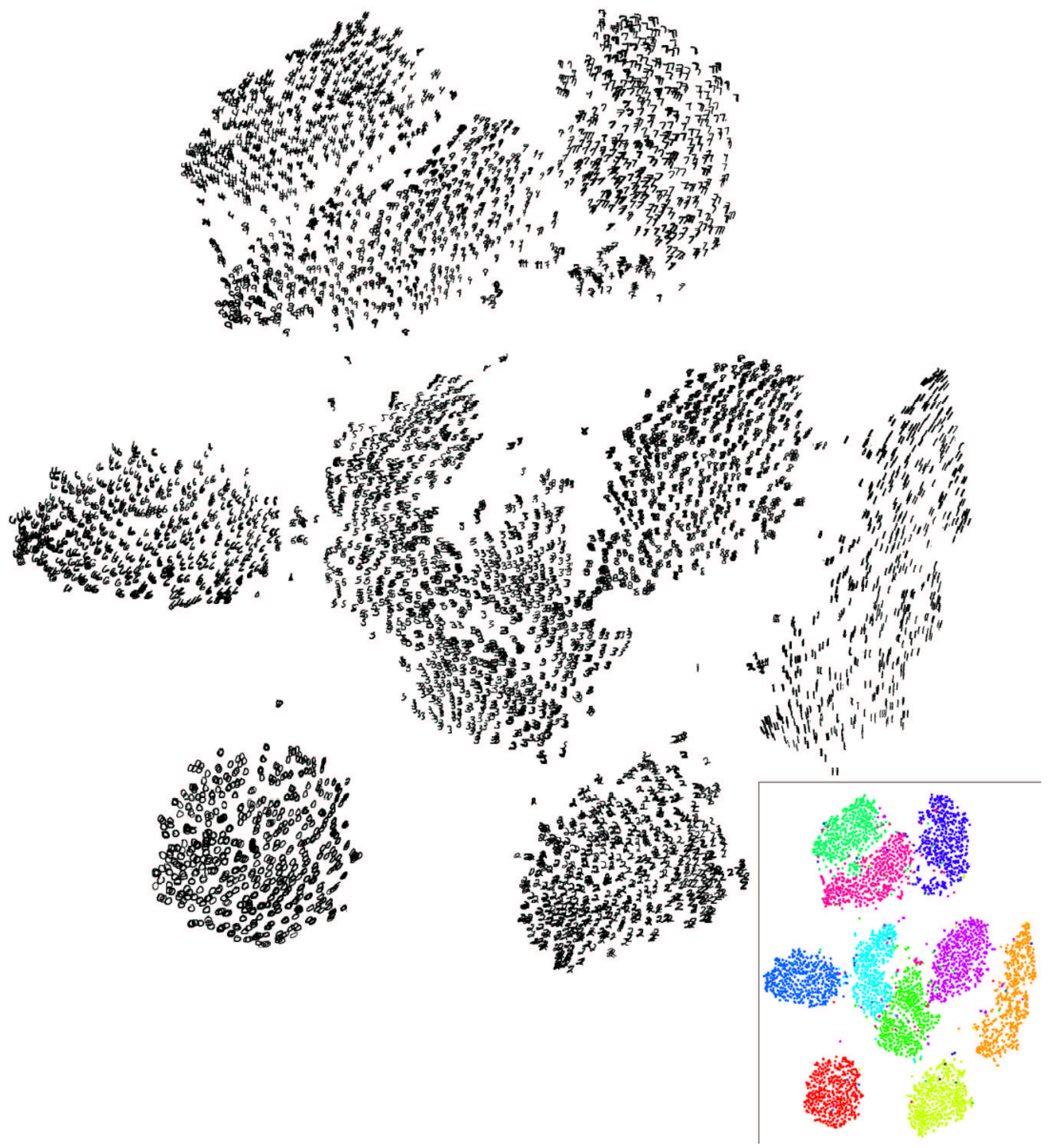
Dimensionality reduction : transformation of data **from a high-dimensional** space **into a low-dimensional** space

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)



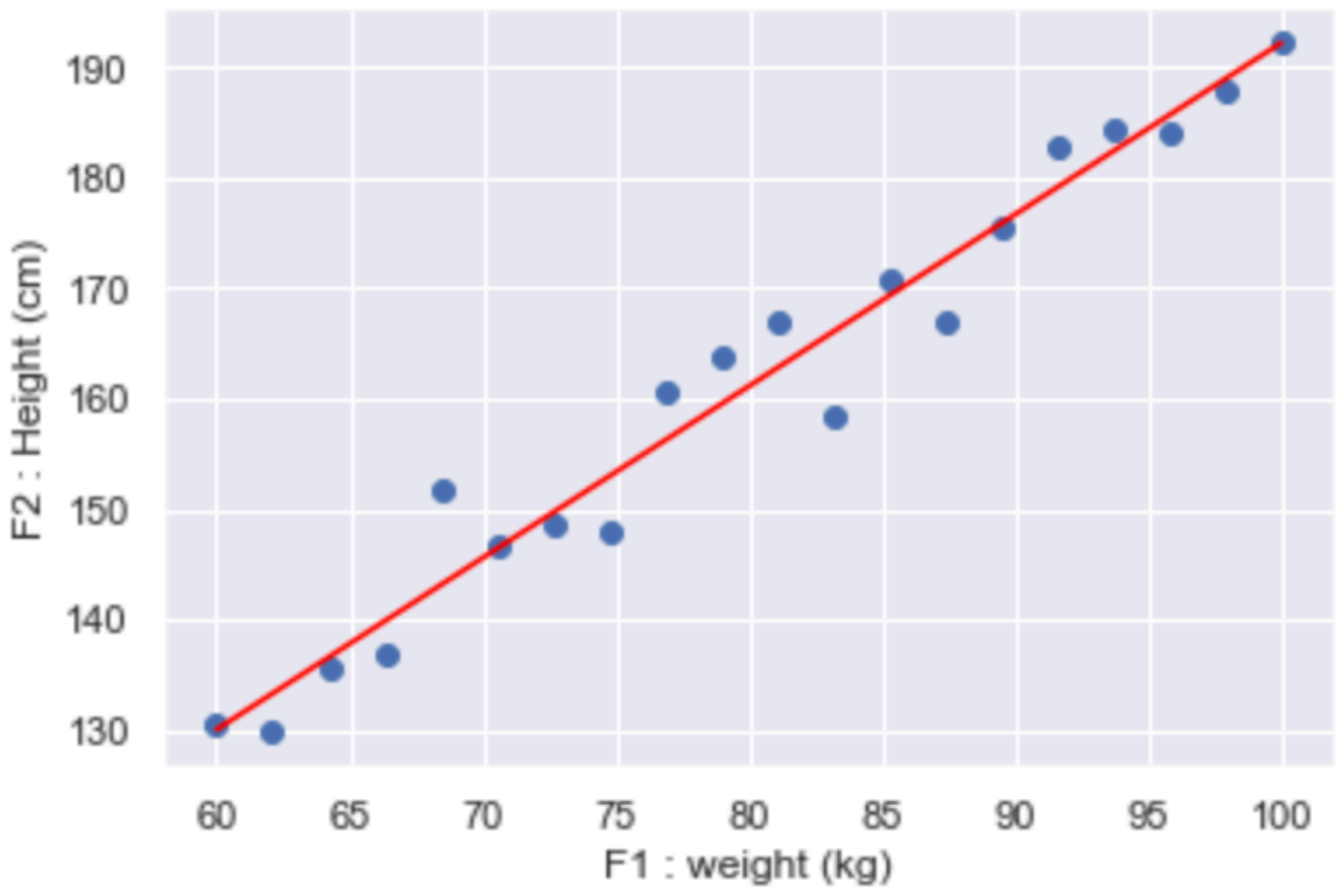
« Visualizing data using t-SNE », JMLR, Laurens et. al, 2008

3. Probabilistic dimensionality reduction

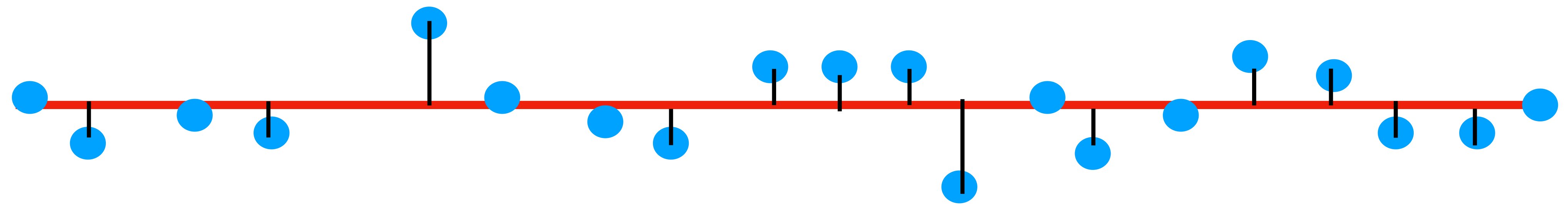
Dimensionality reduction : PCA

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

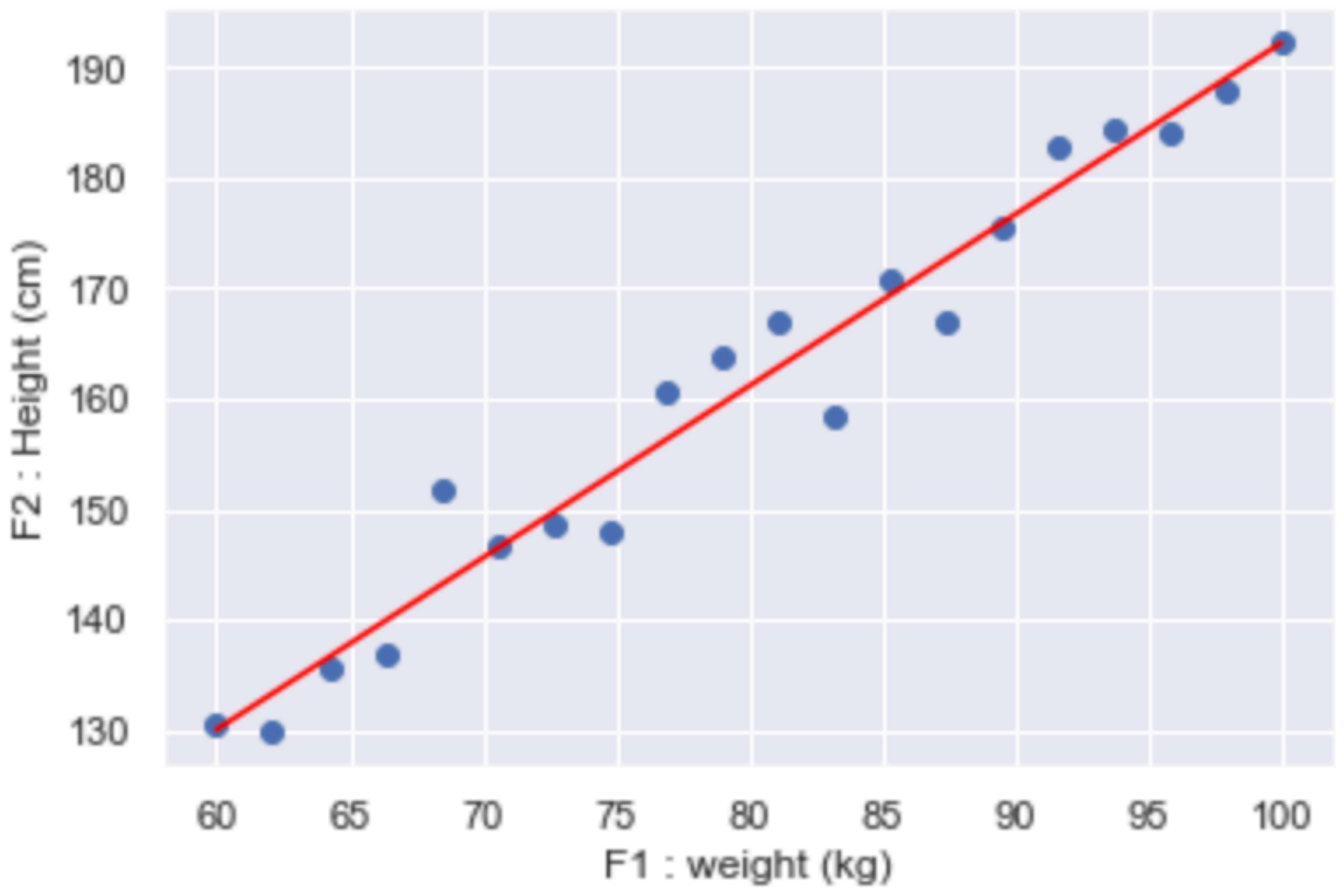


3. Probabilistic dimensionality reduction

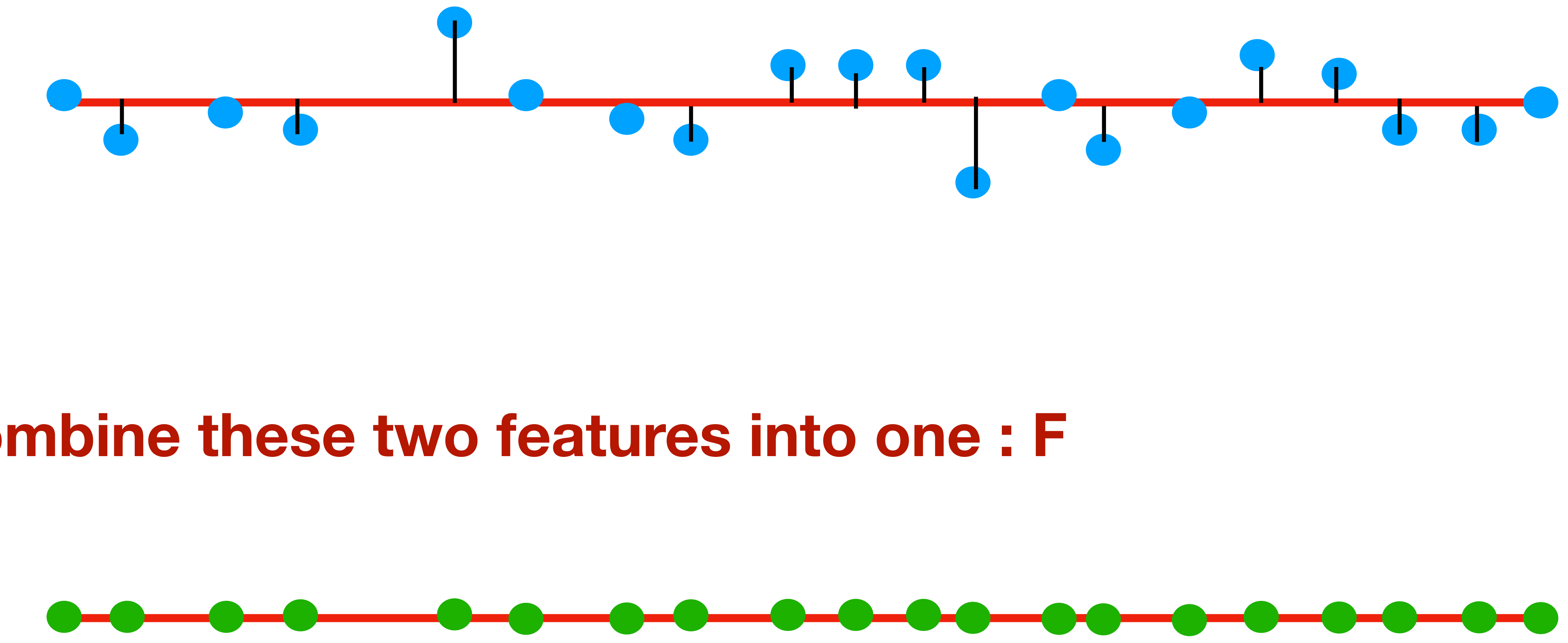
Dimensionality reduction : PCA

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



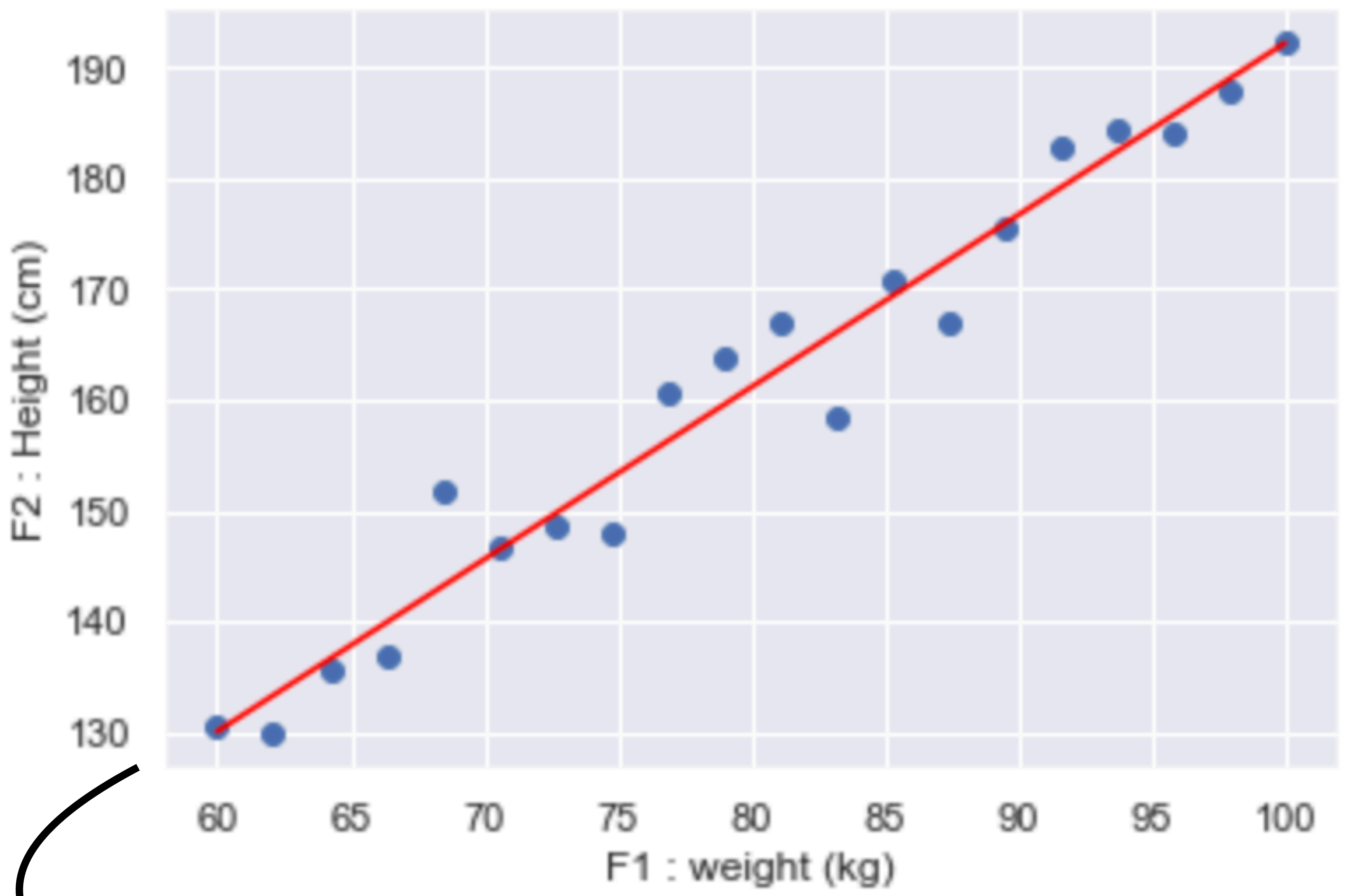
Combine these two features into one : F

3. Probabilistic dimensionality reduction

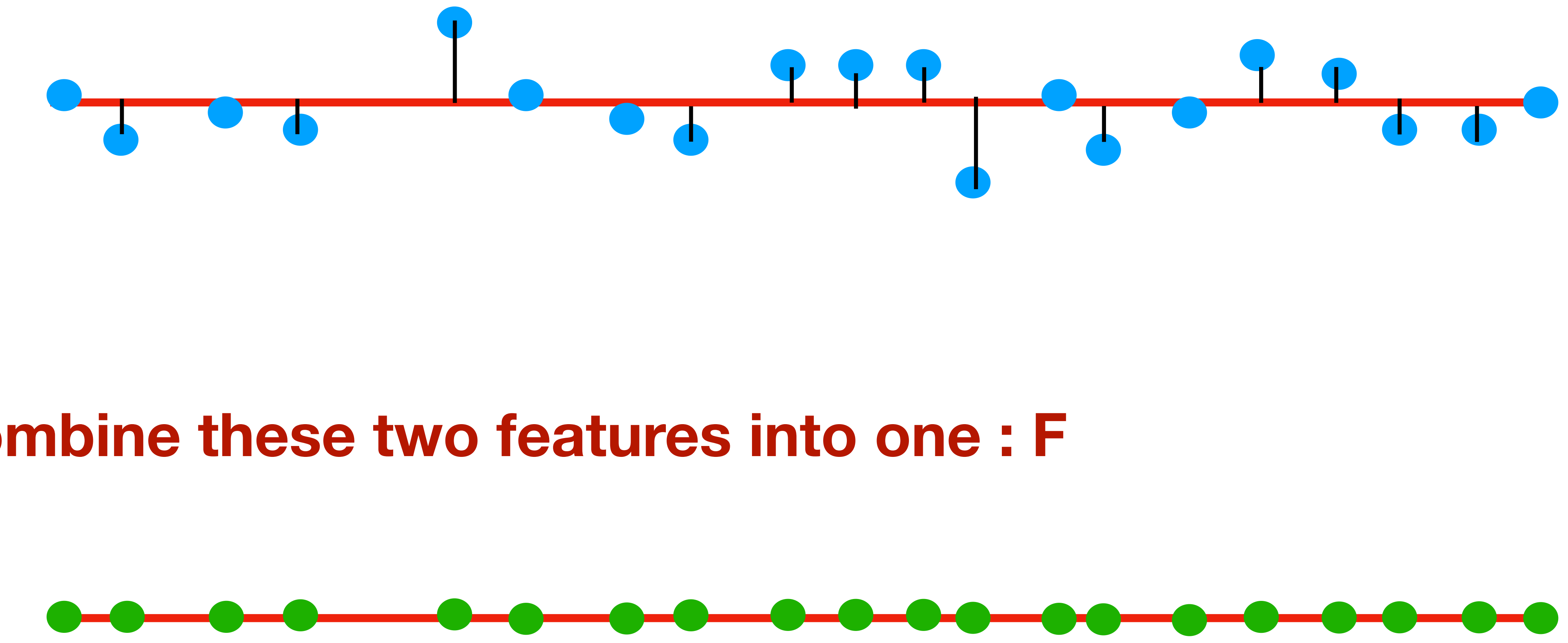
Dimensionality reduction : PCA

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

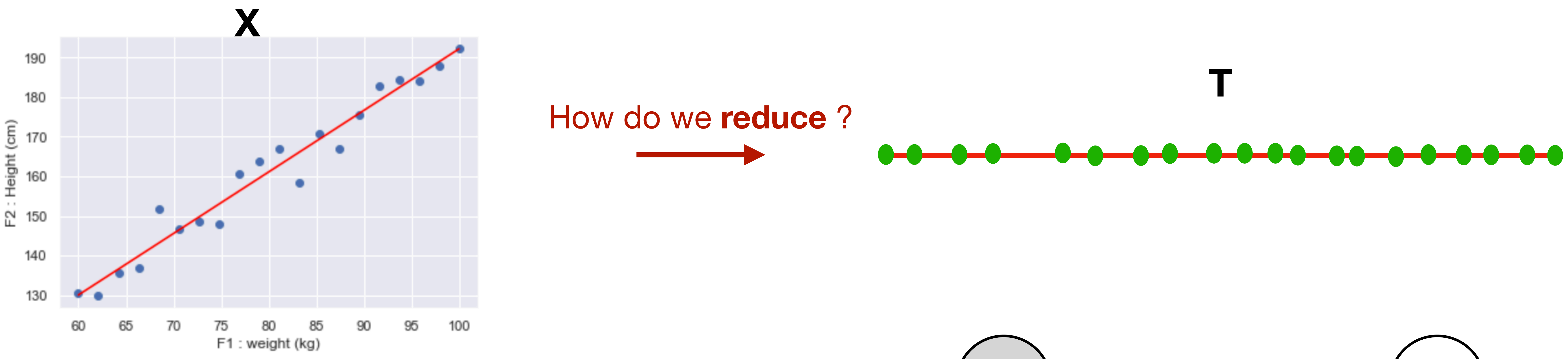
This line corresponds to the eigenvector associated to the greatest eigenvalue of the covariance matrix

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



Probabilistic PCA : a probabilistic point of view of PCA

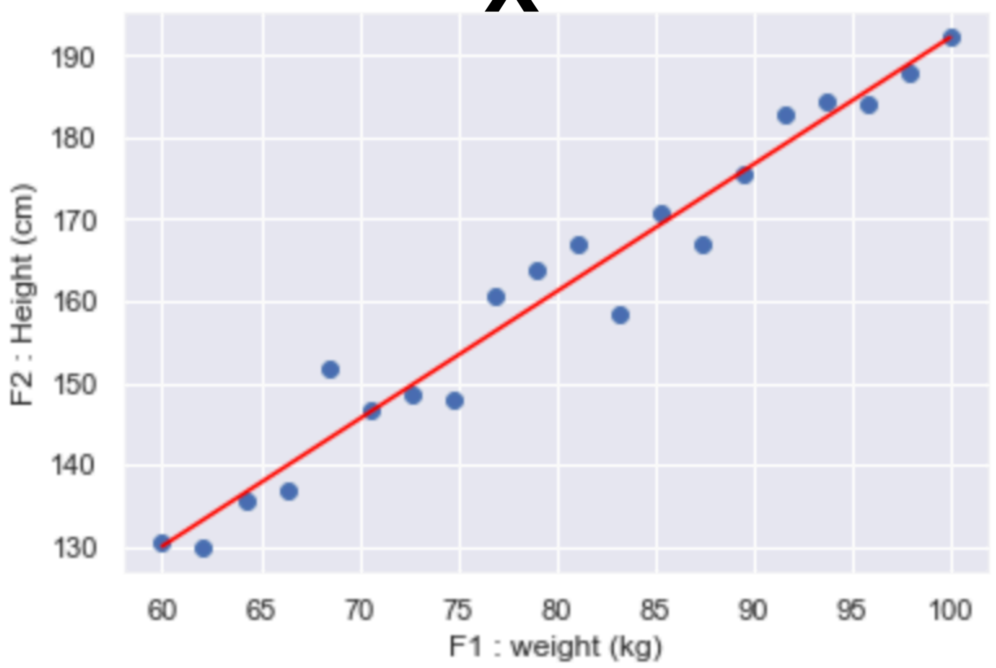
3. Probabilistic dimensionality reduction

Dimensionality reduction : probabilistic PCA (PPCA)

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

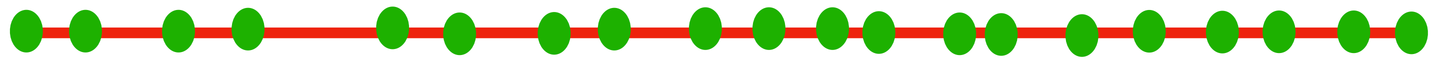
Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data

X



How do we **reduce** ?

T



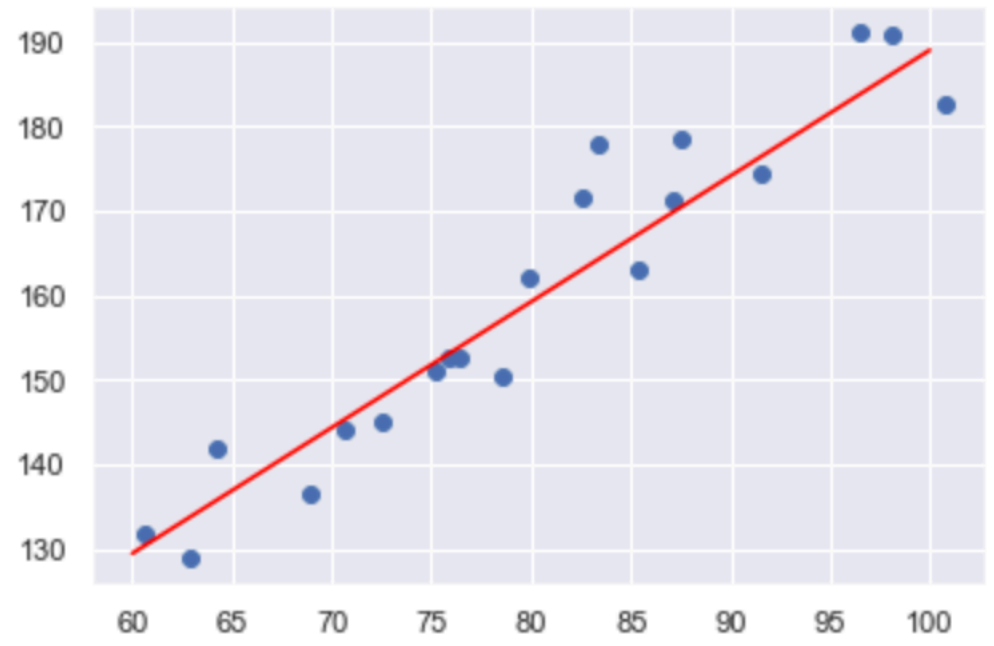
Probabilistic PCA : a probabilistic point of view of PCA

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

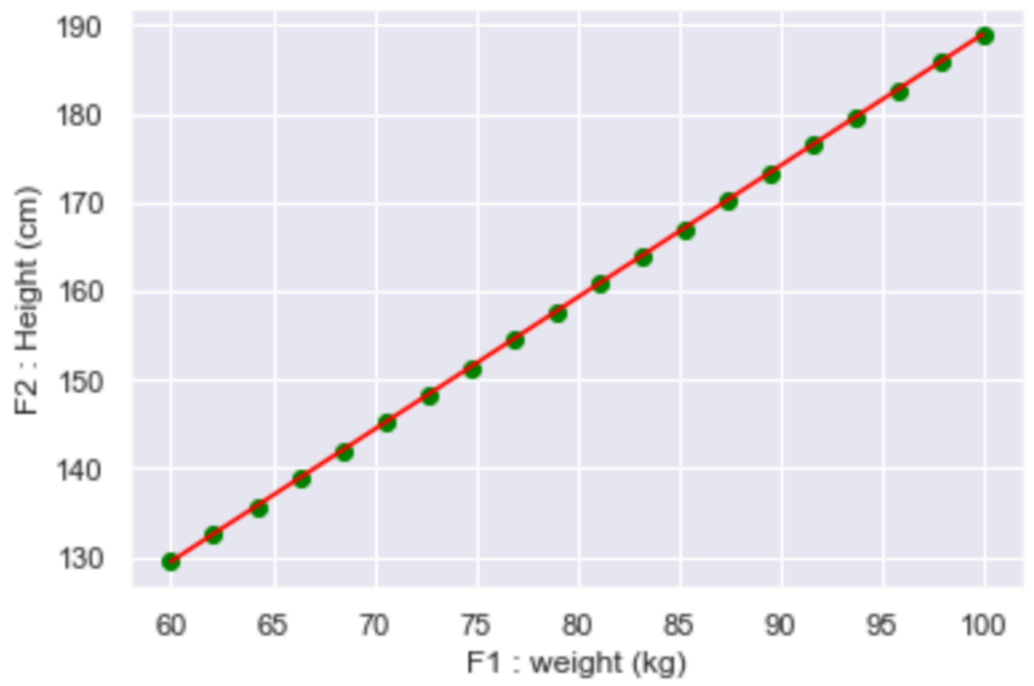
$$x_i = W\, t_i + b$$

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

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



How do we **generate** ?



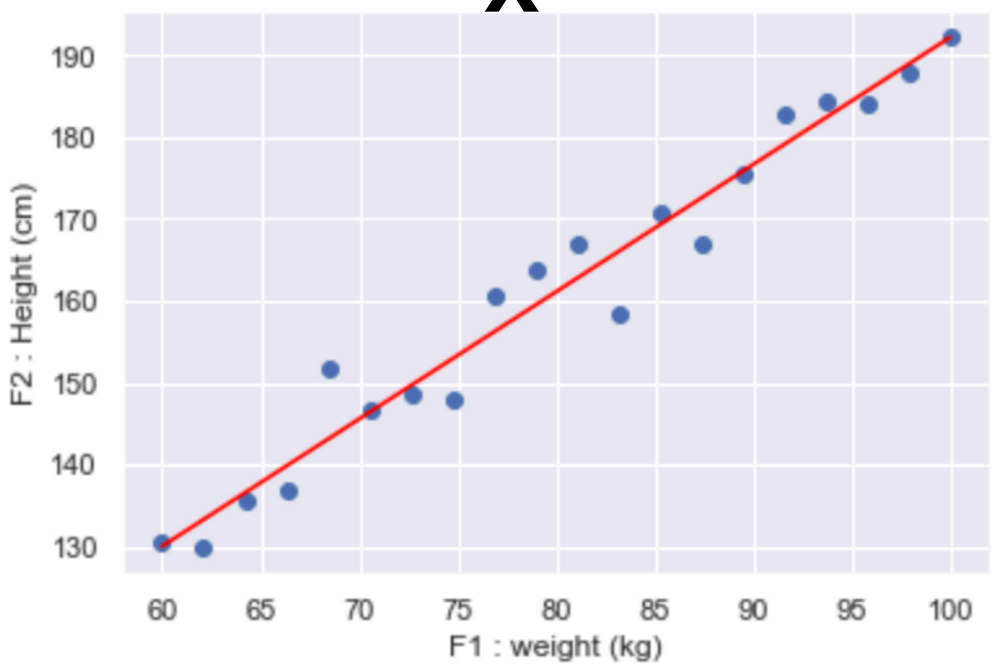
3. Probabilistic dimensionality reduction

Dimensionality reduction : probabilistic PCA (PPCA)

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

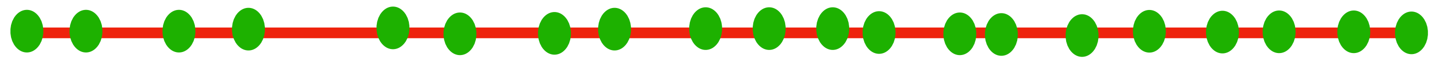
Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data

X



How do we **reduce** ?

T



Probabilistic PCA : a probabilistic point of view of PCA

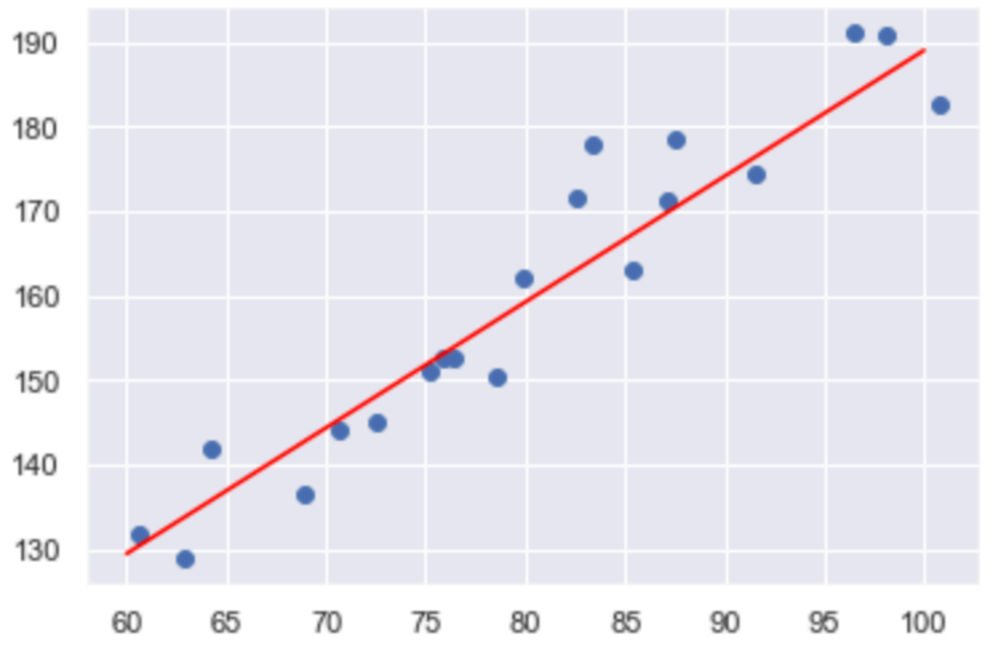
$$p(t_i) = \mathcal{N}(t_i \mid 0,\, I_2)$$

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

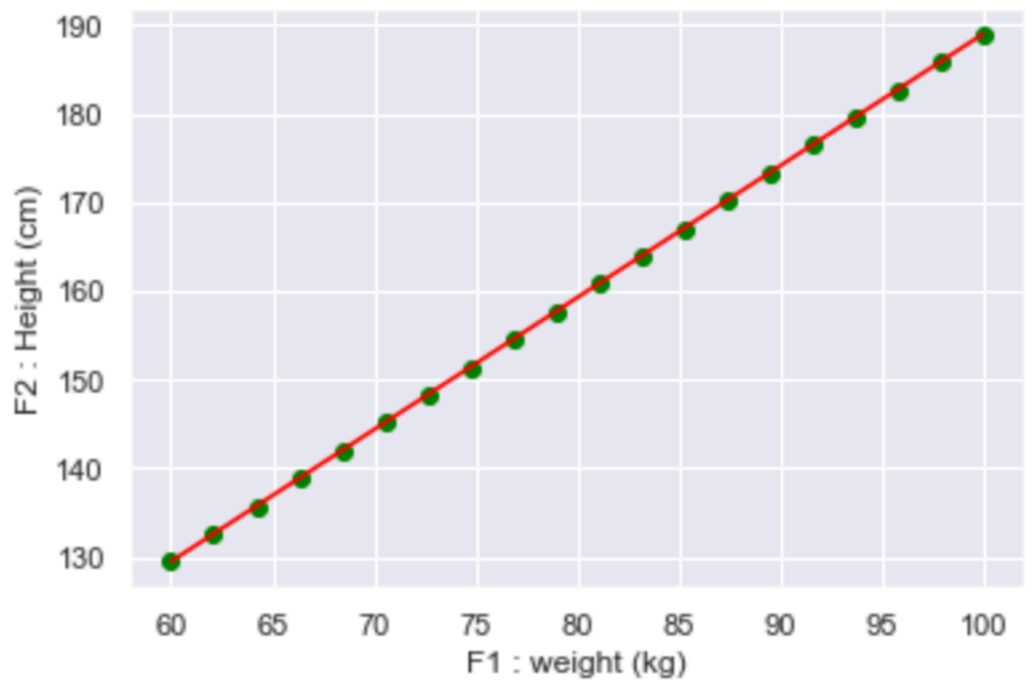
$$x_i = W\, t_i + b + \epsilon_i \text{ with } \epsilon_i \sim \mathcal{N}(0, \Sigma)$$

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

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



How do we **generate** ?

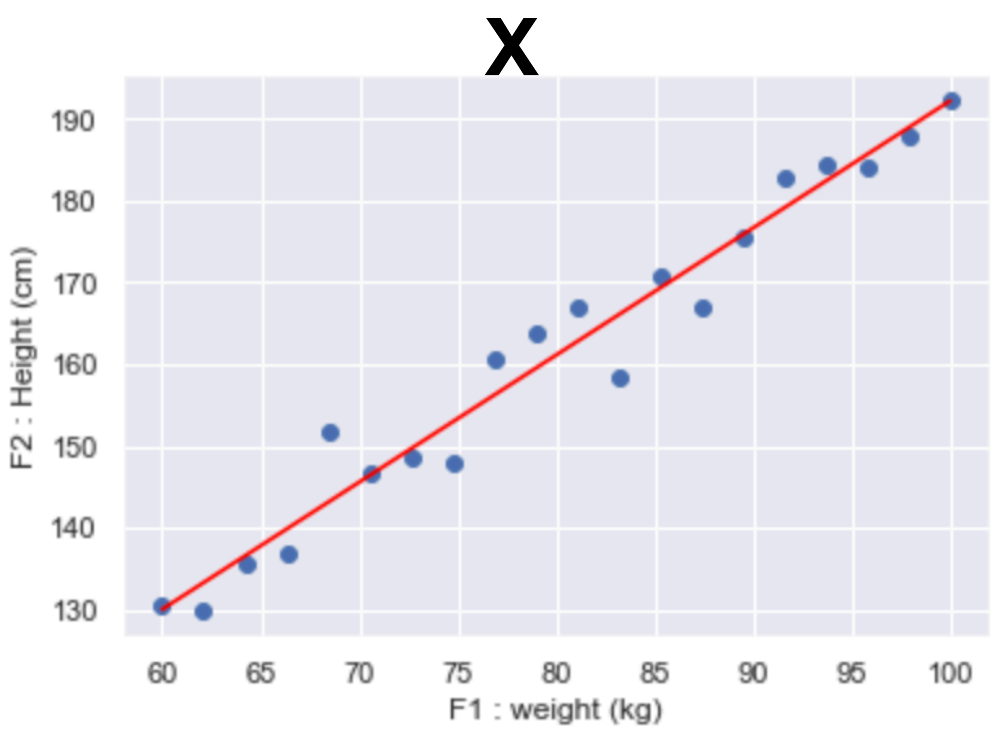


3. Probabilistic dimensionality reduction

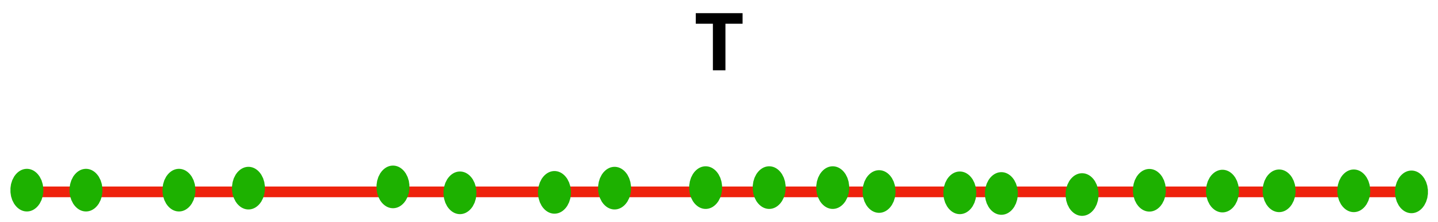
Dimensionality reduction : probabilistic PCA (PPCA)

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** ?



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

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

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

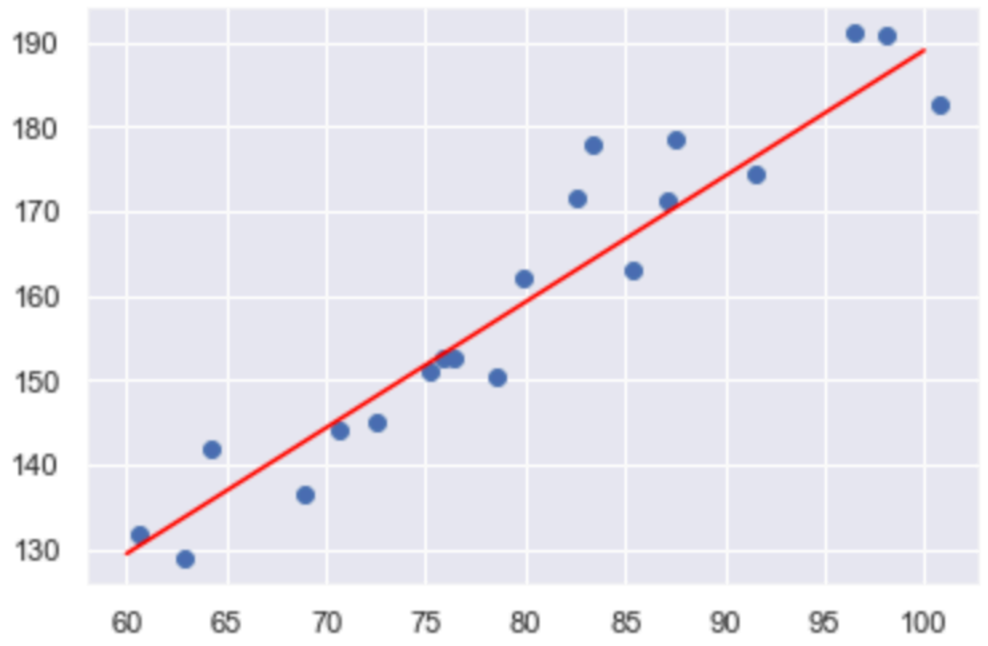
$$p(x_i \,|\, t_i, \theta) = \mathcal{N}(W t_i + b, \Sigma)$$

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

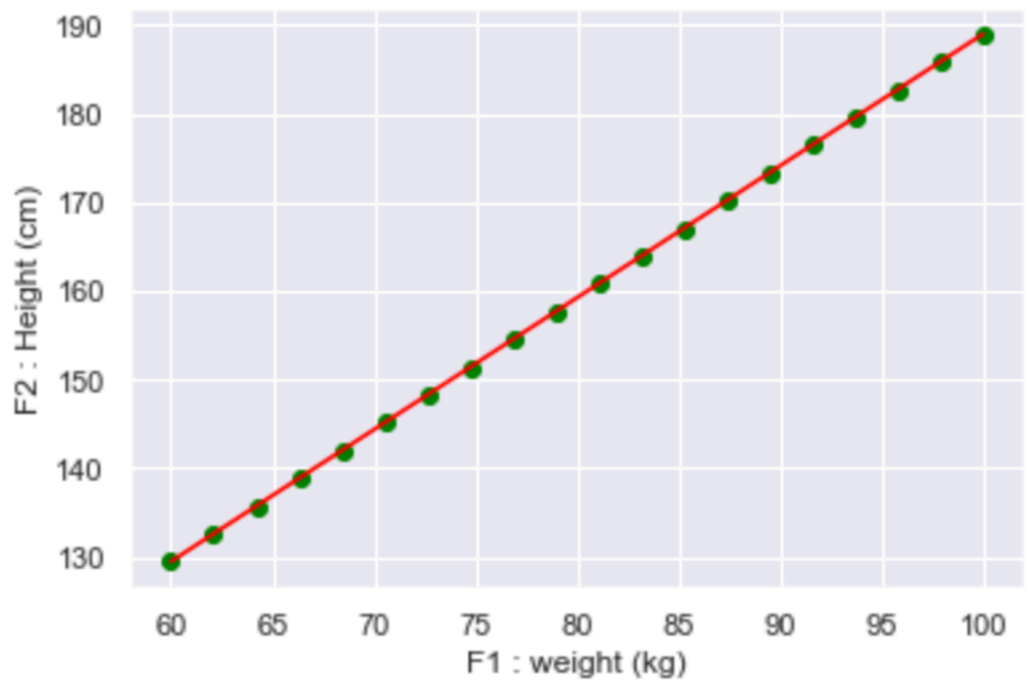
$$= \prod_{i=1,\ldots,n} \int p(x_i \,|\, t_i, \theta) p(t_i) dt_i$$

Normal conjugacy !

Probabilistic PCA : a probabilistic point of view of PCA



How do we **generate** ?



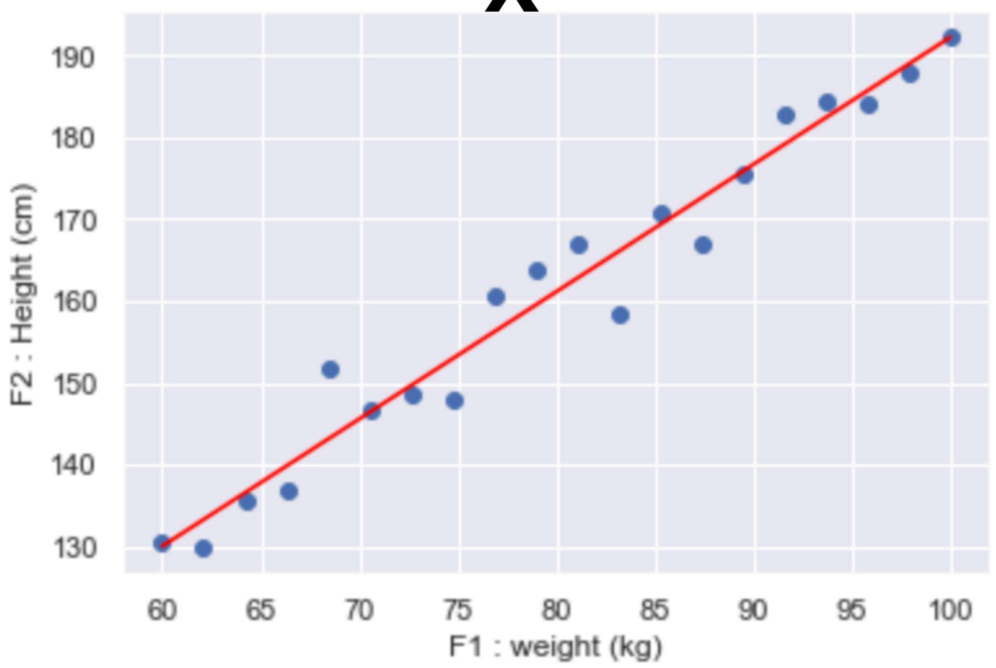
3. Probabilistic dimensionality reduction

Dimensionality reduction : probabilistic PCA (PPCA)

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

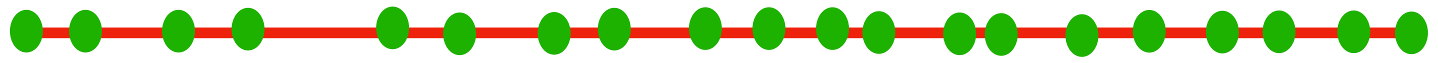
Principal Component Analysis (PCA) : Linear approach to dimensionality reduction : the idea is to linearly project the high-dimensional data into a low-dimensional data

X

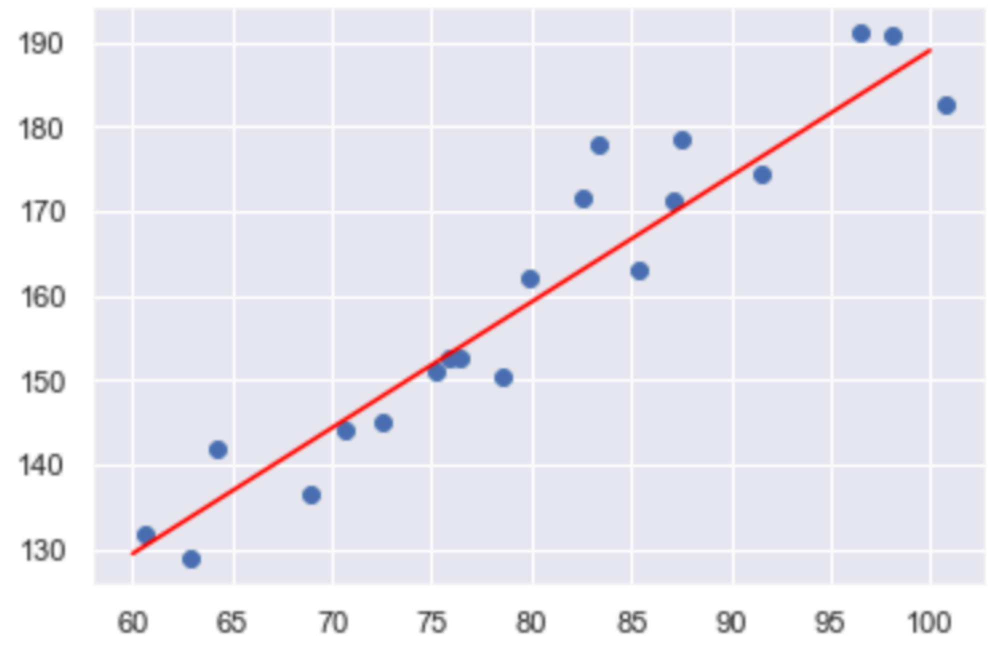


How do we **reduce** ?

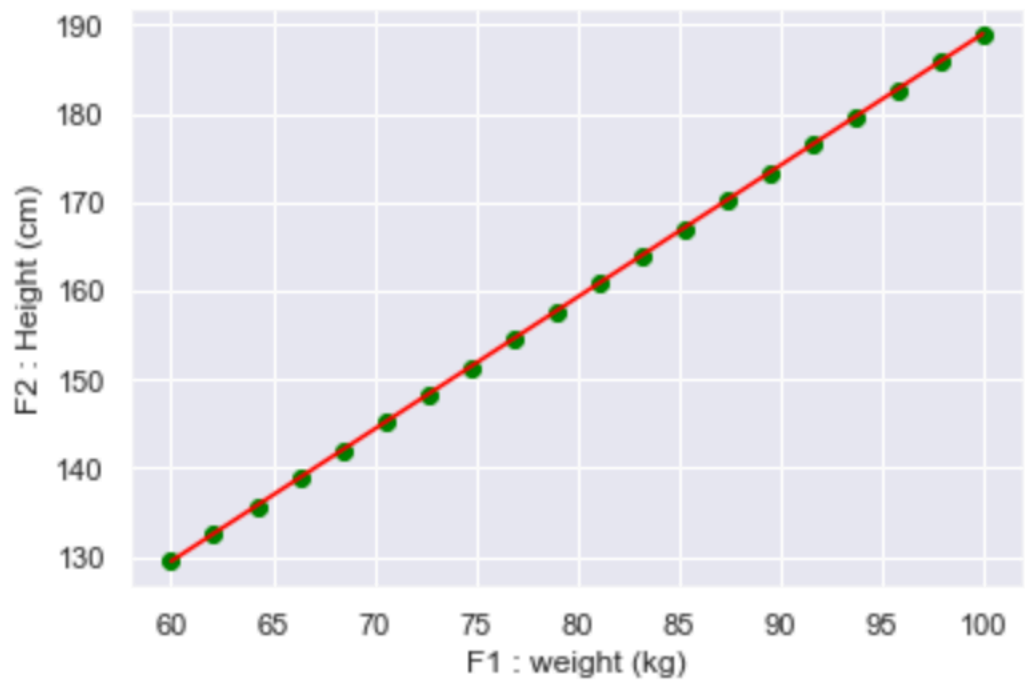
T



Probabilistic PCA : a probabilistic point of view of PCA



How do we **generate** ?



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

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

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

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

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

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

Normal conjugacy !

Easy to do EM here !

3. Probabilistic dimensionality reduction

Dimensionality reduction : probabilistic PCA (PPCA)

Probabilistic PCA : a probabilistic point of view of PCA

$$X \longleftarrow T$$

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 : $\max\limits_{\theta} \leftarrow E_{q(T)} \sum\limits_i \log p(x_i \mid t_i, \theta)\, p(t_i)$

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

$$= \sum_i \log\left(\frac{1}{\text{const}}\right) + \sum_i \underbrace{E_{q(t_i)} \log\left(e^{-\frac{(x - Wt_i - b)^2}{2\sigma^2}}\, e^{-\frac{t_i^2}{2}}\right)}$$

quadratic function on t,
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/

- Master of Science in Artificial Intelligence Systems : **Bayesian Machine Learning** by François HU
    - **Lecture 1** : Bayesian statistics [Lecture]
    - **Lecture 2** : Latent Variable Models and EM-algorithm [Soon available]
    - **Lecture 3** : Variational Inference and intro to NLP [Soon available]
    - **Lecture 4** : Markov Chain Monte Carlo [Soon available]
    - **Lecture 5** : [Oral presentations]
    - **Training session / prerequisite** : Statistics with python [Notebook], [Data]
    - **Practical work 1** : Conjugate distributions [Notebook] [Correction]
    - **Practical work 2** : Probabilistic K-means and probabilistic PCA [Notebook]
    - **Practical work 3** : Topic Modeling with LDA [Soon available]
    - **Practical work 4** : MCMC samples [Soon available]

TODO

! Road map

1 Bayesian statistics

Bayesian perspective :

$$\underbrace{P(\theta|X)}_{\text{Posterior distribution}} = \frac{P(X,\theta)}{P(X)} = \frac{\overbrace{P(X|\theta)}^{\text{Likelihood}} \cdot \overbrace{P(\theta)}^{\text{Prior distribution}}}{\underbrace{P(X)}_{\text{Evidence}}}$$

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 :	Cons :
- exact posterior	- conjugate prior maybe inadequate

2 Latent variable models

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 :	Cons :
- fewer parameters / simpler models - hidden variable sometimes meaningful - clustering / dimensionality reduction	- harder to work with - requires math - only local maximum or saddle point - EM : the posterior of T could be intractable

3 Variational Inference

4 Causal Inference

5 Oral presentation & Extensions