

Introduction to machine learning

Modeling, inference, learning

Simon Leglaive

CentraleSupélec

Today

The key concepts you should be familiar with at the end of this course are the following:

- **Modeling**, or how to define a model that relates the observed data and the latent variables of interest;
- **Inference**, or how to infer the latent variables from the observations;
- **Learning**, or how to estimate the unknown model parameters from the observed data.

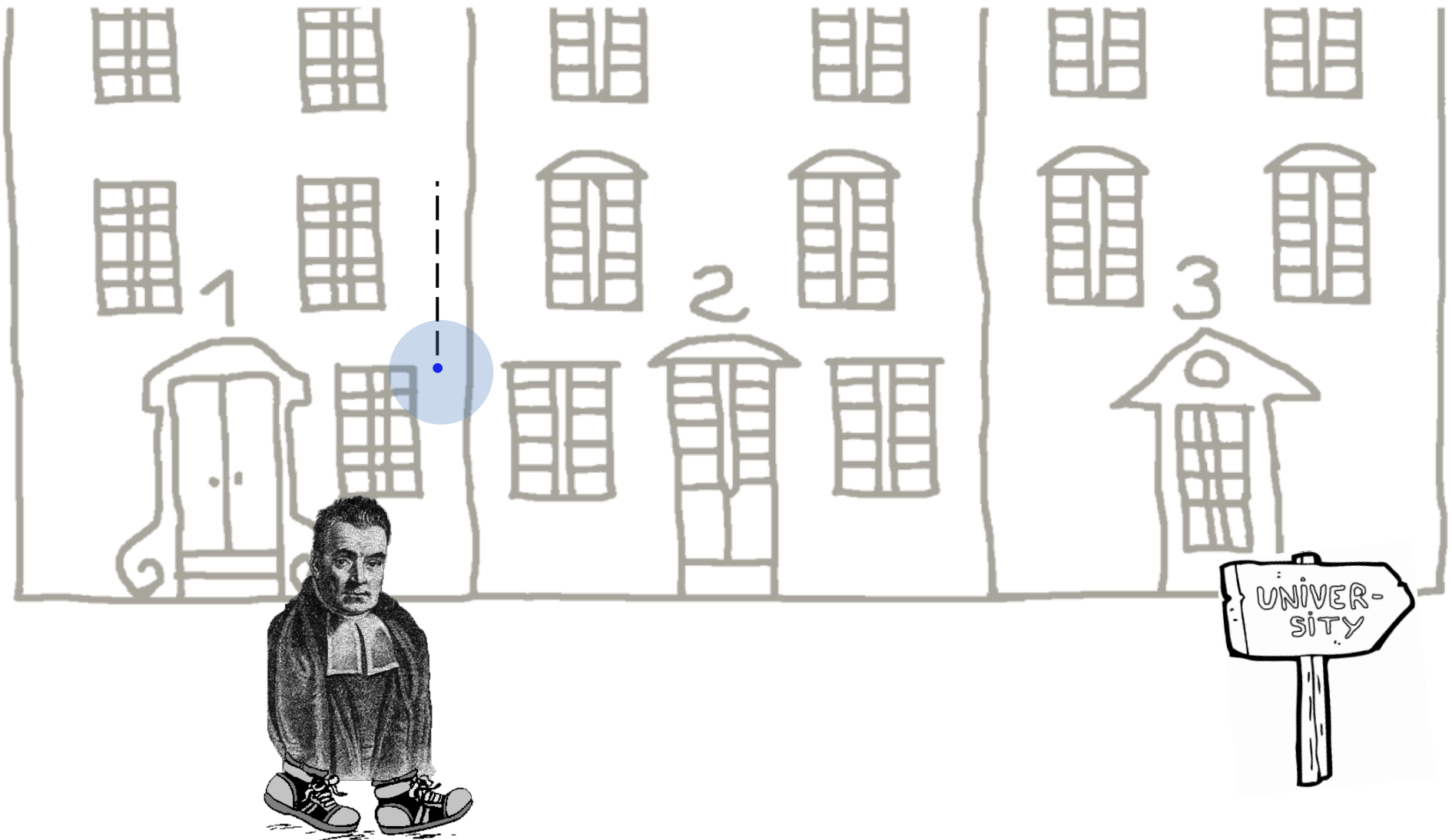
These concepts will be exemplified using the Gaussian mixture model, which will be the focus of the next practical session.

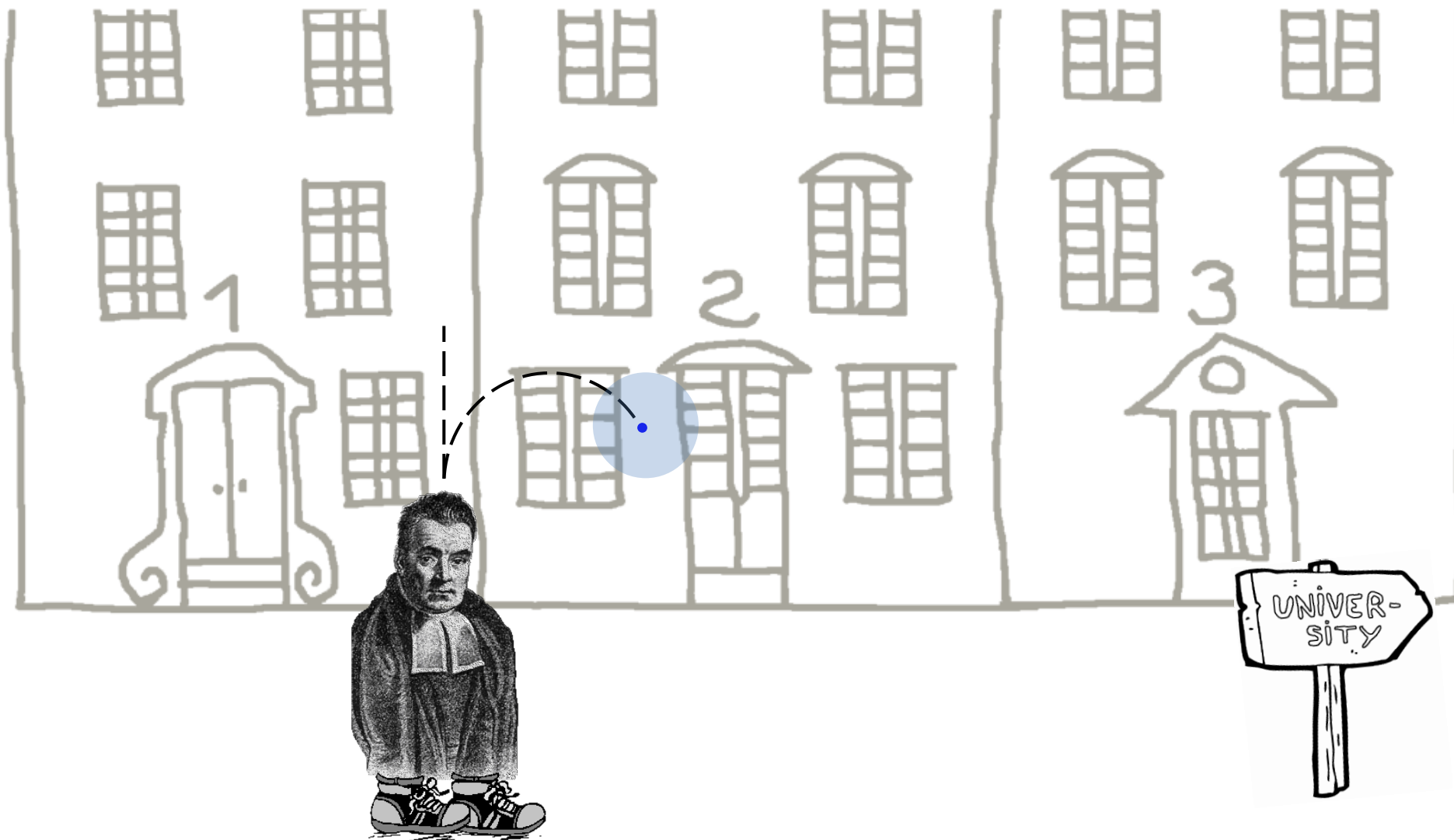
- The deluge of data calls for automated methods of data analysis, which is what machine learning provides.
- Machine learning can be defined as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to perform predictions and/or make decisions (Murphy, 2012).
- Let's start with an introductory example!

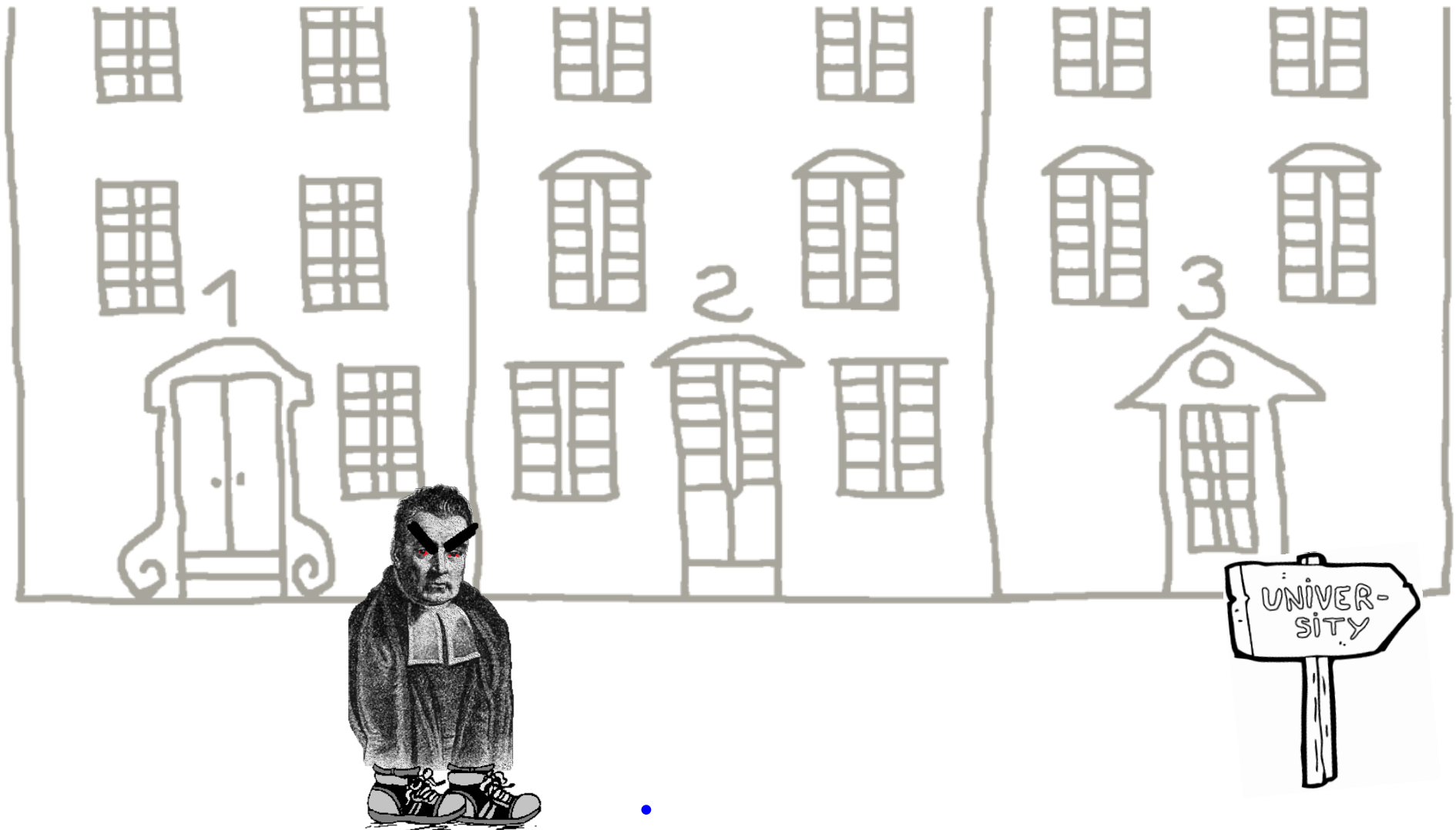
The adventures of Thomas Bayes, episode 1

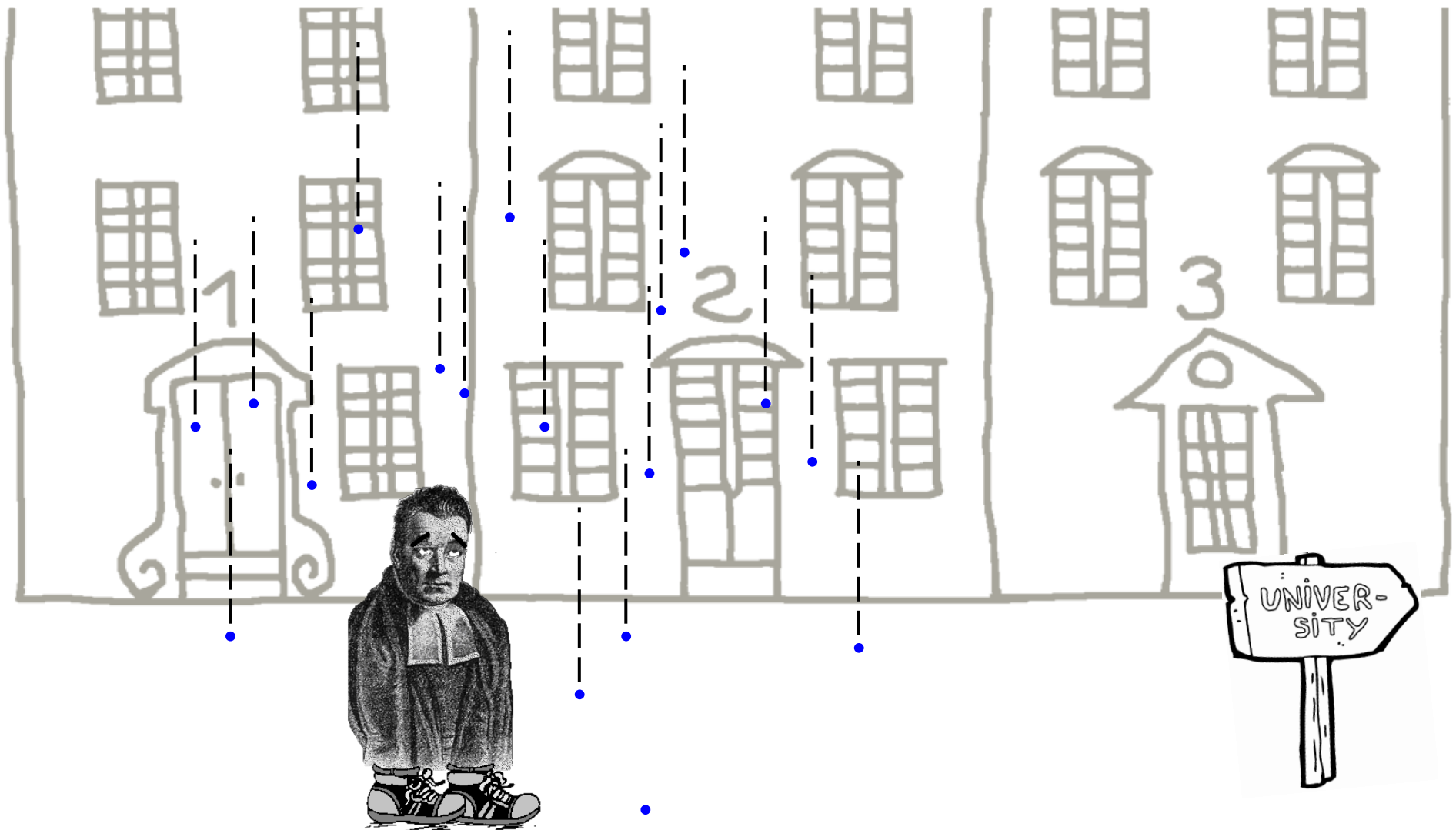
The following example and drawings are adapted from a [tutorial on Bayesian Learning for Signal Processing](#) given by Antoine Deleforge at the LVA/ICA 2015 Summer School.



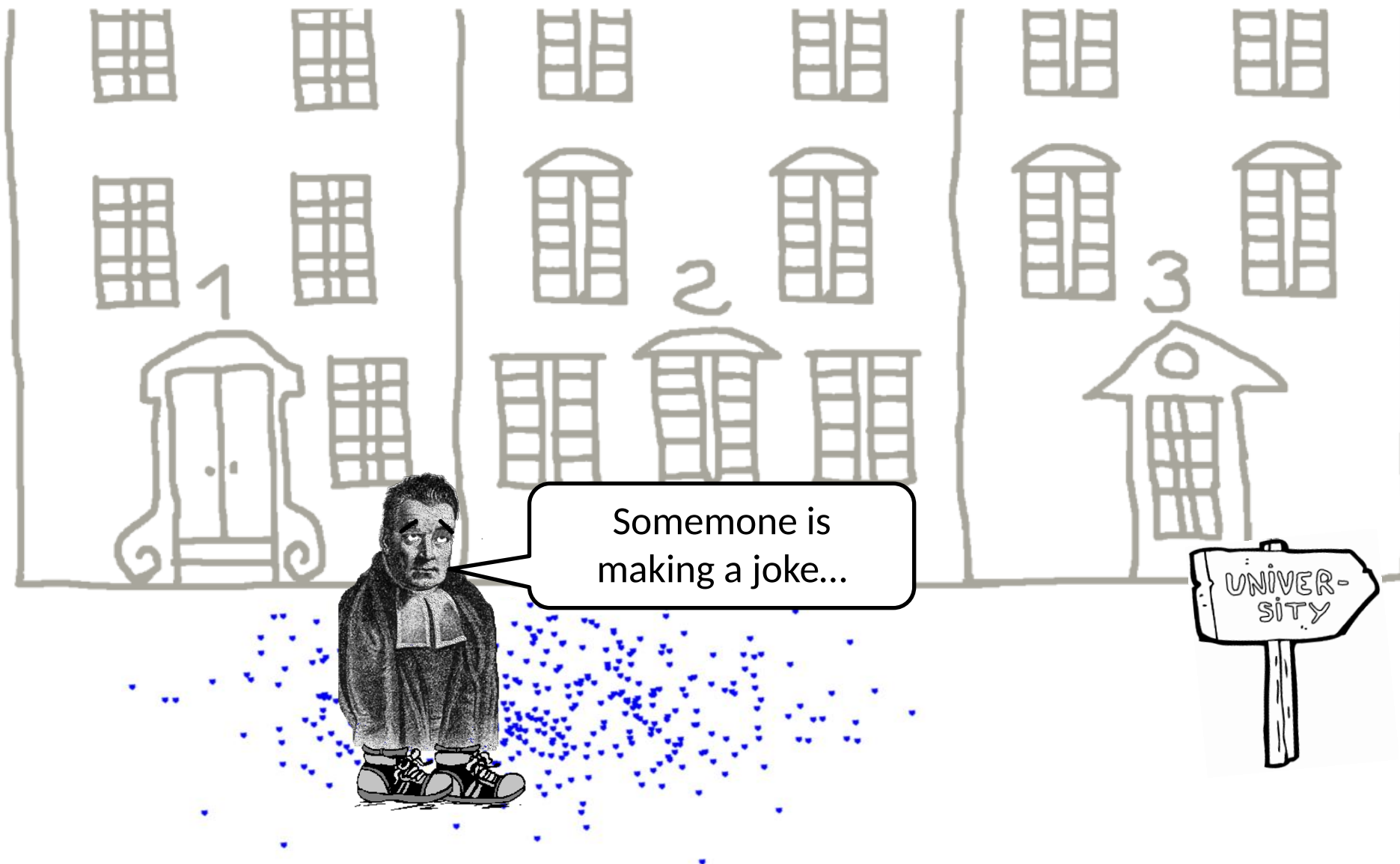






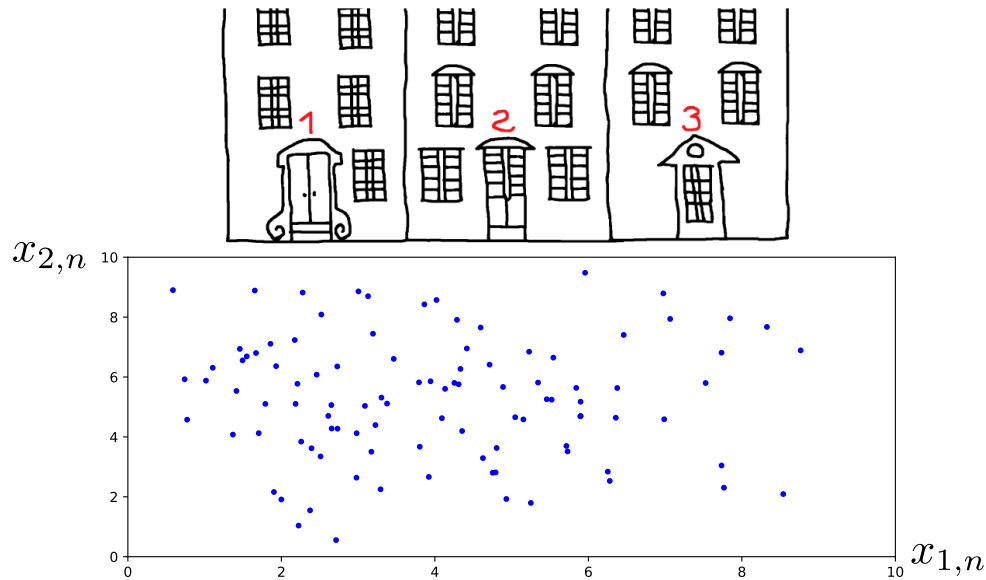








Data / observations



- The dataset \mathcal{D} consists of N observations $\mathbf{x}_i \in \mathbb{R}^D, i = 1, \dots, N$.
- Here, $D = 2$ and \mathbf{x}_i corresponds to the coordinates of the i -th stone on the ground.
- The observations are assumed to be
 1. independent and identically distributed (i.i.d);
 2. generated from an unknown probability distribution $p^*(\mathbf{x})$.

$$\text{We write } \mathcal{D} = \left\{ \mathbf{x}_i \in \mathbb{R}^D \stackrel{i.i.d}{\sim} p^*(\mathbf{x}) \right\}_{i=1}^N.$$

Problem

The problem is to infer a latent variable of interest from the observed data.

Bayes is interested in **inferring** the index of the guilty house, from which the stones were thrown.

This is **the latent variable of interest**, the unknown that we would like to estimate. It is not directly observable, but it is somehow **linked** to the observations.

To solve the problem we need to formalize it.

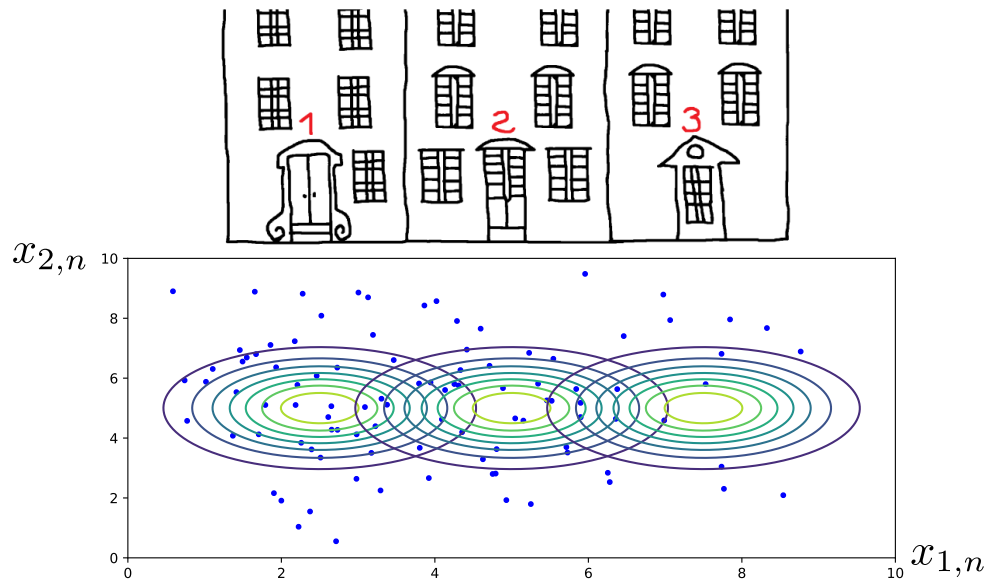
To **formalize the problem**, we need to introduce a discrete variable $z \in \{1, 2, 3\}$ that represents the latent variable and to relate it to the observed data $\mathcal{D} = \{\mathbf{x}_i \in \mathbb{R}^2\}_{i=1}^N$ with a **model**. This model defines the link between what is observed and what is unknown.

Observation (or likelihood) model

The observation model explains how the observations are generated from the latent variable.

Conditionally on z , the observations are **assumed** to be i.i.d according to a Gaussian distribution:

$$p(\mathcal{D} \mid z = k) = p(\mathbf{x}_1, \dots, \mathbf{x}_i \mid z = k) = \prod_{i=1}^N p(\mathbf{x}_i \mid z = k) = \prod_{i=1}^N \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I}).$$



- $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \sigma^2\}$ is a set of parameters assumed to be known and fixed;
- $p(\mathcal{D} \mid z = k)$ is the joint distribution of all the observations and it is called the (conditional) **likelihood**.

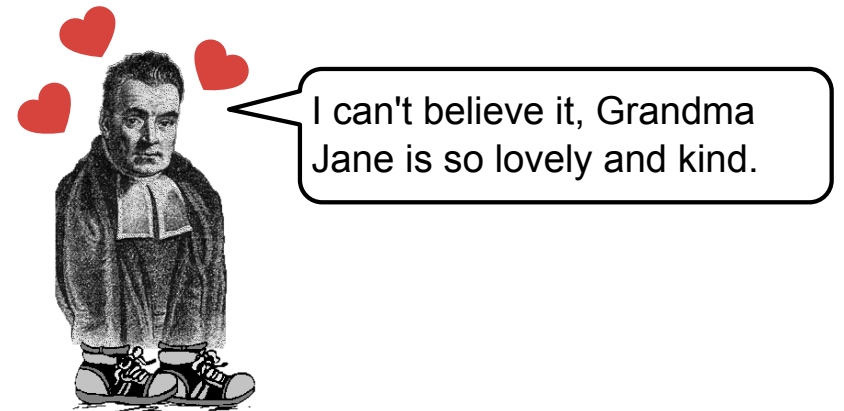
$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the probability density function (pdf) of the **multivariate Gaussian distribution**, where \mathbf{x} is the continuous random vector, $\boldsymbol{\mu}$ the mean vector, and $\boldsymbol{\Sigma}$ the covariance matrix.

Prior model

The prior model encodes prior information / belief / knowledge about the latent variable of interest.

Bayes knows that students, grandma Jane and a family with kids live in the first, second and third house, respectively. So he considers the following prior:

- $\pi_1 := p(z = 1) = 0.3$ (student house);
- $\pi_2 := p(z = 2) = 0.1$ (grandma Jane);
- $\pi_3 := p(z = 3) = 0.6$ (family with kids).



What prior could Bayes choose if he did not know the occupants of the different houses?

For the discrete random variable z , $p(z = k)$ denotes the probability that it is equal to k .

$A := B$ reads "A is defined to be B".

Inference

In the most general case, inference consists in computing or approximating the posterior distribution of the latent variable of interest.

This is achieved by using Bayes' theorem.

$$p(z = k | \mathcal{D}) = \frac{p(\mathcal{D} | z = k)p(z = k)}{p(\mathcal{D})} \quad \text{(using Bayes theorem)}$$

$$= \frac{p(\mathcal{D} | z = k)p(z = k)}{\sum_{k=1}^3 p(\mathcal{D}, z = k)} \quad \text{(using the sum rule)}$$

$$= \frac{p(\mathcal{D} | z = k)p(z = k)}{\sum_{k=1}^3 p(\mathcal{D} | z = k)p(z = k)} \quad \text{(using the product rule)}$$

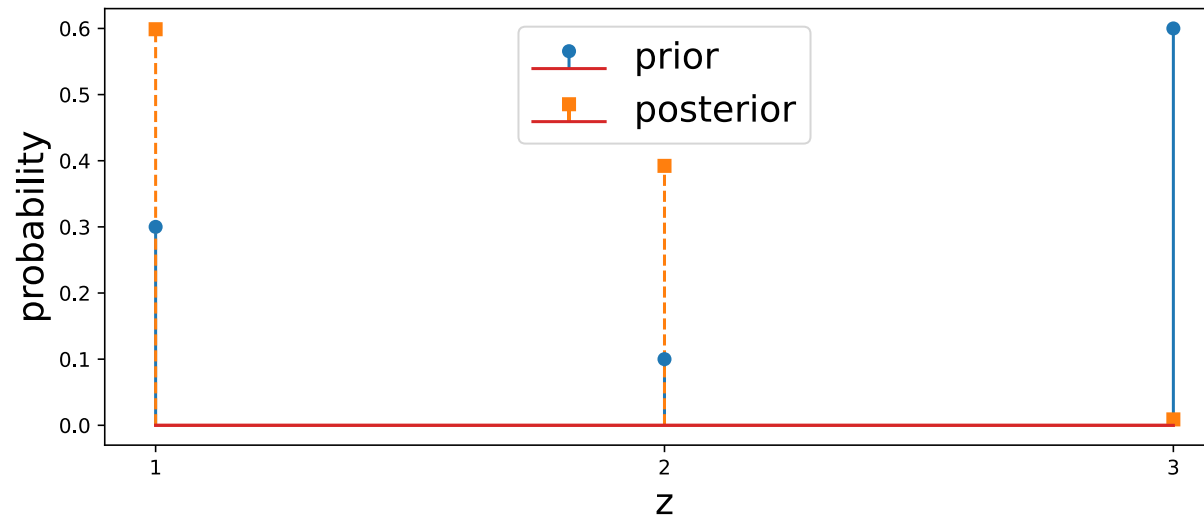
$$= \frac{p(z = k) \prod_{i=1}^N p(\mathbf{x}_i | z = k)}{\sum_{k=1}^3 p(z = k) \prod_{i=1}^N p(\mathbf{x}_i | z = k)} \quad \text{(using the i.i.d assumption)}$$

$$= \frac{\pi_k \prod_{i=1}^N \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})}{\sum_{k=1}^3 \pi_k \prod_{i=1}^N \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})} \quad \text{(using the prior and observation model)}$$

$$p(z = k | \mathcal{D}) = \frac{\pi_k \prod_{i=1}^N \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})}{\sum_{k=1}^3 \pi_k \prod_{i=1}^N \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})}$$

The posterior combines the information from the prior and from the observations. It updates the prior using the observations, through the Bayes' theorem.

We have access to all the quantities necessary to compute the posterior distribution.



Point estimate

We are often interested in computing a point estimate of the latent variable of interest from its posterior distribution.

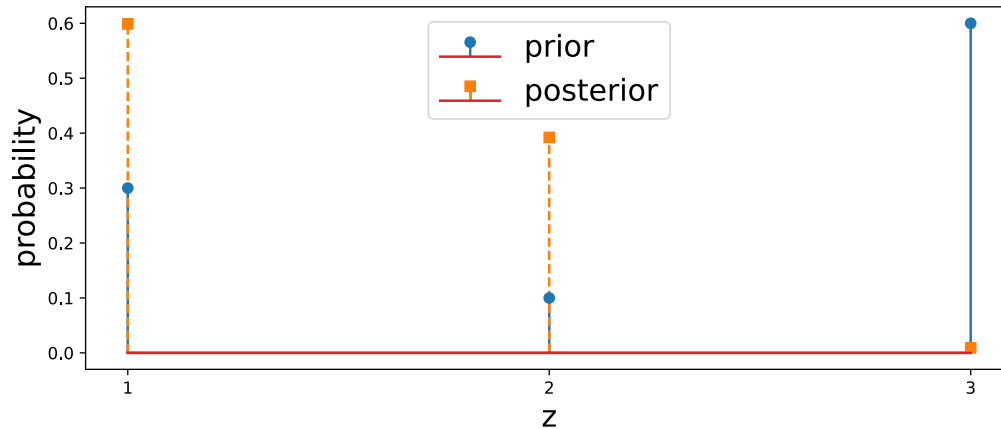
- The posterior contains all the information about the latent variable we care about, but it does not directly tell Bayes which house is the guilty one.
- From the posterior $p(z = k \mid \mathcal{D})$, $k \in \{1, 2, 3\}$, Bayes needs to **make a decision** about the guilty house.
- This is achieved by computing a **point estimate** $\hat{z} \in \{1, 2, 3\}$, and the posterior probability $p(z = \hat{z} \mid \mathcal{D})$ indicates how **confident** (or equivalently uncertain) Bayes is about this decision.

In estimation theory and decision theory, the point estimate is called the **Bayes estimator**. It is defined as the minimizer of a posterior expected loss (the expectation of a loss function taken with respect to the posterior distribution). Various loss functions can be defined, leading to different estimates.

A natural choice here is to take the **maximum a posteriori** (MAP) estimate:

$$\hat{z}_{\text{MAP}} = \arg \max_{k \in \{1,2,3\}} p(z = k \mid \mathcal{D}) = 1.$$

The students are (estimated) guilty!



These pranksters will hear from me at the Uni!

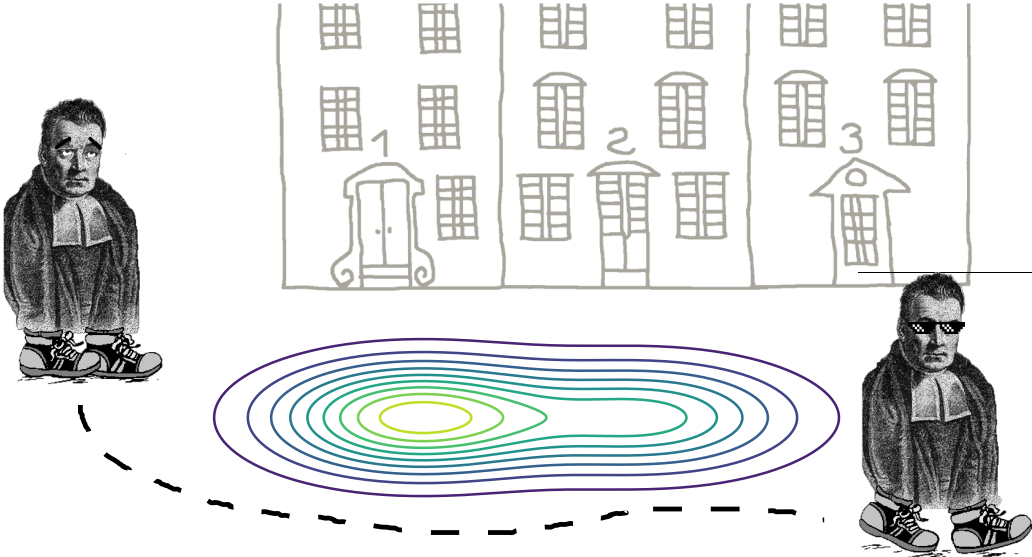
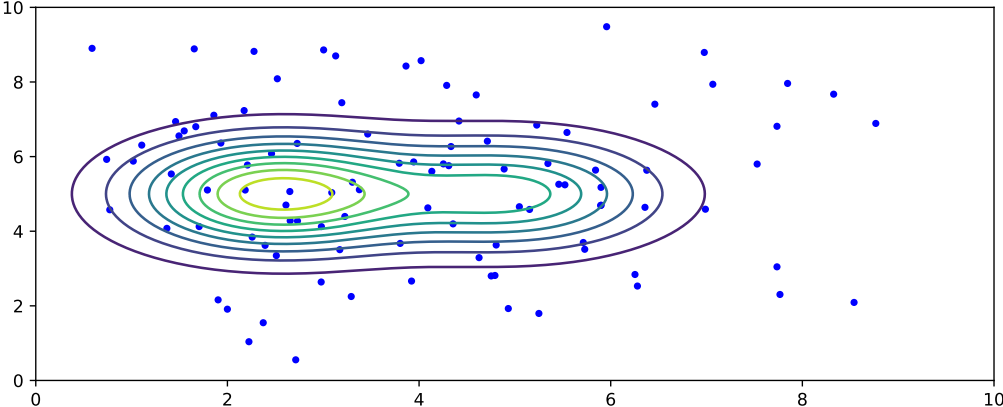
Prediction / generation of new data

We can also predict new data given the already observed ones using the predictive posterior.

$$\begin{aligned} p(\mathbf{x}_{\text{new}} \mid \mathcal{D}) &= \sum_{k=1}^3 p(\mathbf{x}_{\text{new}}, z = k \mid \mathcal{D}) && \text{(using the sum rule)} \\ &= \sum_{k=1}^3 p(\mathbf{x}_{\text{new}} \mid z = k, \mathcal{D}) p(z = k \mid \mathcal{D}) && \text{(using the product rule)} \\ &= \sum_{k=1}^3 p(\mathbf{x}_{\text{new}} \mid z = k) p(z = k \mid \mathcal{D}) && \text{(using the independence assumption)} \\ &= \sum_{k=1}^3 \mathcal{N}(\mathbf{x}_{\text{new}}; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I}) p(z = k \mid \mathcal{D}) && \text{(using the Gaussian observation model)} \\ &= \mathbb{E}_{p(z=k|\mathcal{D})} [\mathcal{N}(\mathbf{x}_{\text{new}}; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})] && \text{(using the definition of the expectation)} \end{aligned}$$

The predictive posterior is an average of the observation model weighted by the posterior probabilities of z .

The next day, Bayes goes to the university armed with his **predictive posterior**:

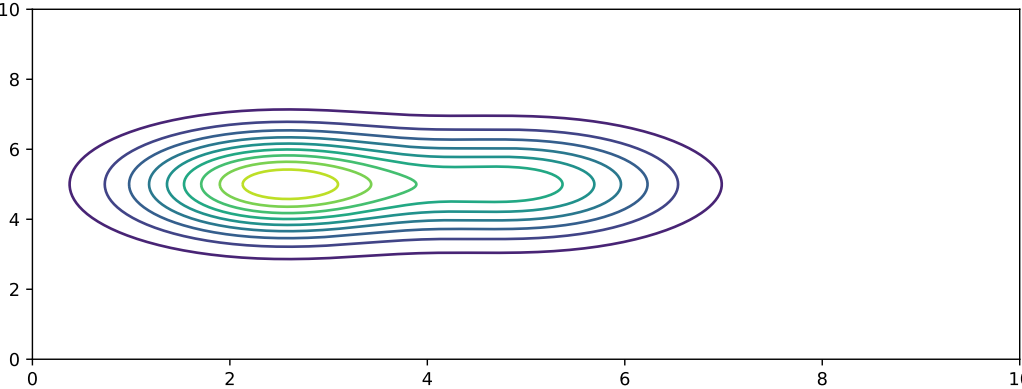
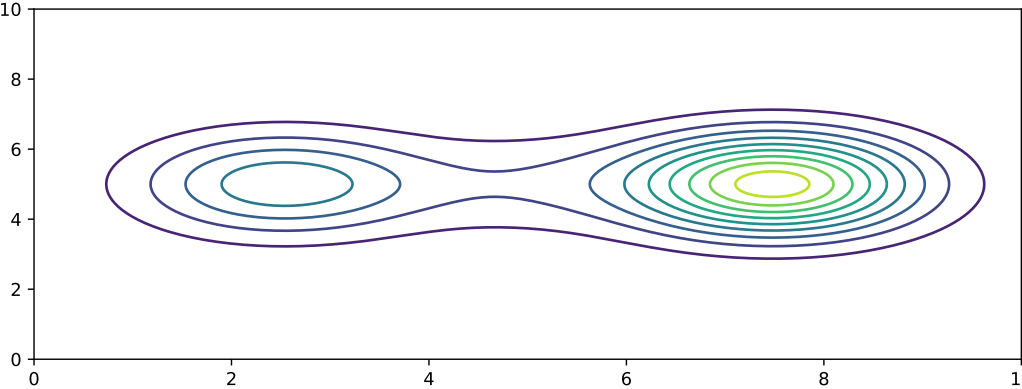


We can also compute the **predictive prior**, which tells us what we would predict given no observations. This is useful to check if the prior distribution does capture our prior beliefs.

Predictive prior



Predictive posterior



$$\mathbb{E}_{p(z=k)} [\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})]$$

$$\mathbb{E}_{p(z=k|\mathcal{D})} [\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})]$$

1st house: students; 2nd house: grandma; 3rd house: kids

Wrap-up

Modeling, inference, and learning

Starting point

We started from the general problem of inferring some latent information from observations in a dataset

$$\mathcal{D} = \left\{ \mathbf{x}_i \stackrel{i.i.d}{\sim} p^*(\mathbf{x}) \right\}_{i=1}^N.$$

Modeling

We formalized the problem by **defining a model that links the observed and the latent variables**.

Following a **generative** approach, this was achieved by defining their **joint distribution**:

$$p(\mathbf{x}, z; \theta) = p(\mathbf{x} \mid z; \theta_x) p(z; \theta_z),$$

where $\theta = \theta_x \cup \theta_z$ and

- $p(\mathbf{x} \mid z; \theta_x)$ is the (conditional) **likelihood** that defines how observations are generated from the latent variable. It depends on deterministic parameters θ_x (mean vectors and variance in Bayes' adventures);
- $p(z; \theta_z)$ is the **prior** that encodes the prior belief and uncertainty about the latent variable of interest. It depends on deterministic parameters θ_z (the prior probabilities in Bayes' adventures);

By defining the prior and the likelihood models we are making assumptions about the generative process of the observed data.

As all observations are assumed to be i.i.d, we drop the index n of \mathbf{x}_i .

For a discrete (resp. continuous) random variable z , $p(z; \theta_z)$ denotes its **probability mass function** (resp. **probability density function**).

By marginalizing the unobserved latent variable in the joint distribution $p(\mathbf{x}, z; \theta)$ we obtain the **marginal likelihood**:

$$p(\mathbf{x}; \theta) = \begin{cases} \int_{\mathcal{Z}} p(\mathbf{x} | z; \theta_x) p(z; \theta_z) dz & \text{if } z \in \mathcal{Z} \text{ is continuous;} \\ \sum_{k \in \mathcal{Z}} p(\mathbf{x} | z = k; \theta_x) p(z = k; \theta_z) & \text{if } z \in \mathcal{Z} \text{ is discrete.} \end{cases}$$

The marginal likelihood $p(\mathbf{x}; \theta)$ is a model of the distribution $p^*(\mathbf{x})$ that is assumed to have generated the observations in the dataset.

Inference

Inference consists in computing the posterior distribution of the latent variable, which summarizes our knowledge on z once we have observed \mathbf{x} .

- Using **Bayes' theorem**, the posterior distribution writes:

$$p(z | \mathbf{x}; \theta) = \frac{p(\mathbf{x} | z; \theta_x)p(\mathbf{z}; \theta_z)}{p(\mathbf{x}; \theta)}.$$

- The process of inference will often require us to use the posterior to answer various questions.

Point estimate

- $p(z \mid \mathbf{x}; \theta)$ encodes all our knowledge about z after observing the data, but it does not directly provide an "estimate" of the z . From the posterior, we need to choose a single value \hat{z} to serve as a **point estimate** of z . In Bayesian statistics, this is a **decision**, and in different contexts we might want to select different point estimates.
- To take the decision, we need to introduce a **loss function** $l(\hat{z}, z)$ which tells us "how bad" would \hat{z} be if the "true value" of the latent variable was z . The decision is then taken by minimizing the **posterior expected loss**:

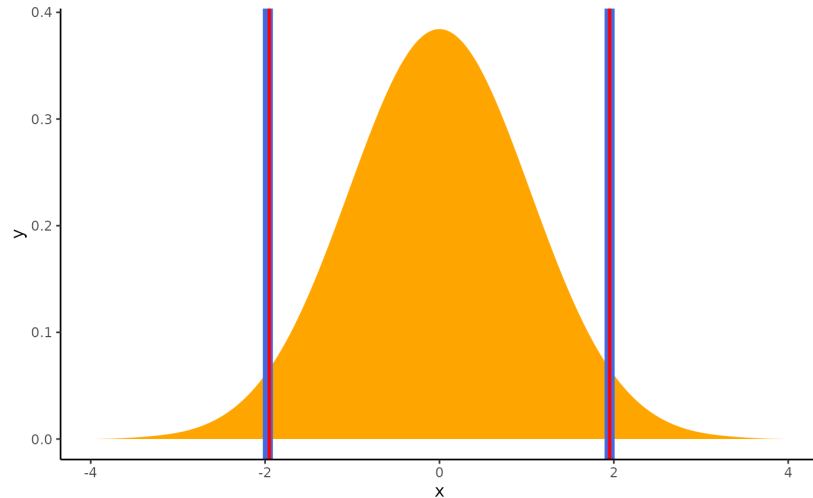
$$\mathcal{L}(\hat{z}) = \mathbb{E}_{p(z|\mathbf{x};\theta)} [l(\hat{z}, z)].$$

- For example, if we consider a continuous latent variable and the squared error loss $l(\hat{z}, z) = (\hat{z} - z)^2$, we obtain the **posterior mean** estimate:

$$\hat{z}_{MSE} = \arg \min_{\hat{z}} \mathbb{E}_{p(z|\mathbf{x};\theta)} [(\hat{z} - z)^2] = \mathbb{E}_{p(z|\mathbf{x};\theta)} [z].$$

Uncertainty

- Importantly, the posterior distribution encodes **uncertainty** about the latent variable of interest. Indeed, we inferred a full probability distribution and did not simply compute a point estimate.
- Quantifying uncertainty when making predictions is important for critical applications such as in medicine, autonomous driving, etc.
- Uncertainty can be quantified using a **credible interval**, which is just an interval within which the latent variable value falls with a particular probability.



$[a, b]$ is the 95% credible interval for the continuous latent variable z if

$$\int_a^b p(z \mid \mathbf{x}; \theta) dz = 0.95.$$

Prediction / generation of new data

- Predictive prior

"Averaging" the likelihood over the prior:

$$p(\mathbf{x}_{\text{new}}; \theta) = \mathbb{E}_{p(z; \theta_z)} [p(\mathbf{x}_{\text{new}} | z; \theta_x)].$$

- Predictive posterior

"Averaging" the likelihood over the posterior:

$$p(\mathbf{x}_{\text{new}} | \mathbf{x}; \theta) = \mathbb{E}_{p(z|\mathbf{x};\theta)} [p(\mathbf{x}_{\text{new}} | z; \theta_x)].$$

Wait, what about **learning**, as in machine **learning**?

Learning

In the adventures of Thomas Bayes, we ended-up with the following decision rule:

$$\hat{z} = \arg \max_{k \in \{1,2,3\}} p(z = k \mid \mathcal{D}; \theta) = \arg \max_{k \in \{1,2,3\}} \frac{\pi_k \prod_{i=1}^N \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})}{\sum_{k=1}^3 \pi_k \prod_{i=1}^N \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \sigma^2 \mathbf{I})}$$

This is a function of:

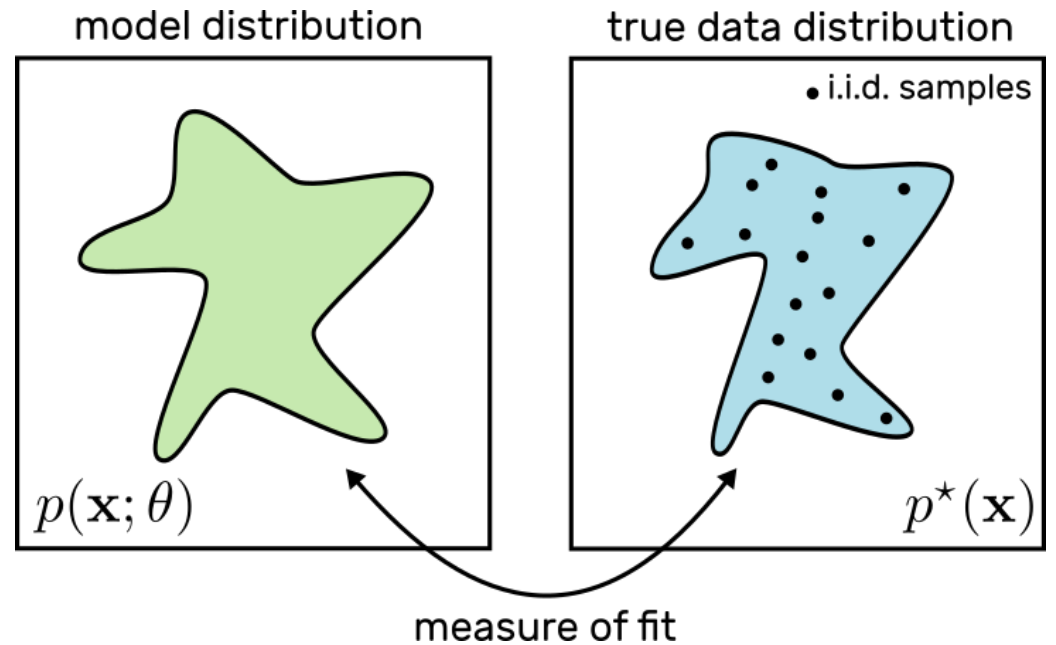
- the input **data** in $\mathcal{D} = \left\{ \mathbf{x}_i \stackrel{i.i.d}{\sim} p^*(\mathbf{x}) \right\}_{i=1}^N$;
- the **model parameters** $\theta = \left\{ \sigma^2, \{ \boldsymbol{\mu}_k, \pi_k \}_{k=1}^3 \right\}$, which were assumed to be known and fixed.

Learning is the process to automatically estimate the model parameters from the data.

Many models in machine learning can be studied from a probabilistic perspective, where learning consists in estimating the parameters θ that make the **model** distribution $p(\mathbf{x}; \theta)$ as close as possible to the true data distribution $p^*(\mathbf{x})$, given a **dataset** of i.i.d observations and a **measure of fit**.

The three main ingredients to formalize learning in probabilistic machine learning are

- A model distribution $p(\mathbf{x}; \theta)$, which may or may not involve latent variables;
- A dataset $\mathcal{D} = \left\{ \mathbf{x}_i \stackrel{i.i.d}{\sim} p^*(\mathbf{x}) \right\}_{i=1}^N$;
- A measure of fit between $p(\mathbf{x}; \theta)$ and $p^*(\mathbf{x})$, seen as a function of θ .



KL divergence and maximum likelihood

- A popular choice is to take the Kullback-Leibler (KL) divergence as a measure of fit:

$$D_{\text{KL}}(p \parallel q) = \mathbb{E}_p[\ln(p) - \ln(q)] \geq 0,$$

with equality if and only if $p = q$ and $D_{\text{KL}}(p \parallel q) \neq D_{\text{KL}}(q \parallel p)$.

- Then **learning consists in solving the following optimization problem:**

$$\min_{\theta} \left\{ D_{\text{KL}}(p^*(\mathbf{x}) \parallel p(\mathbf{x}; \theta)) = \mathbb{E}_{p^*(\mathbf{x})}[\ln p^*(\mathbf{x}) - \ln p(\mathbf{x}; \theta)] \right\} \Leftrightarrow \max_{\theta} \mathbb{E}_{p^*(\mathbf{x})}[\ln p(\mathbf{x}; \theta)].$$

- The difficulty is that we do not know the true data distribution $p^*(\mathbf{x})$, which prevents us from computing the expectation analytically.

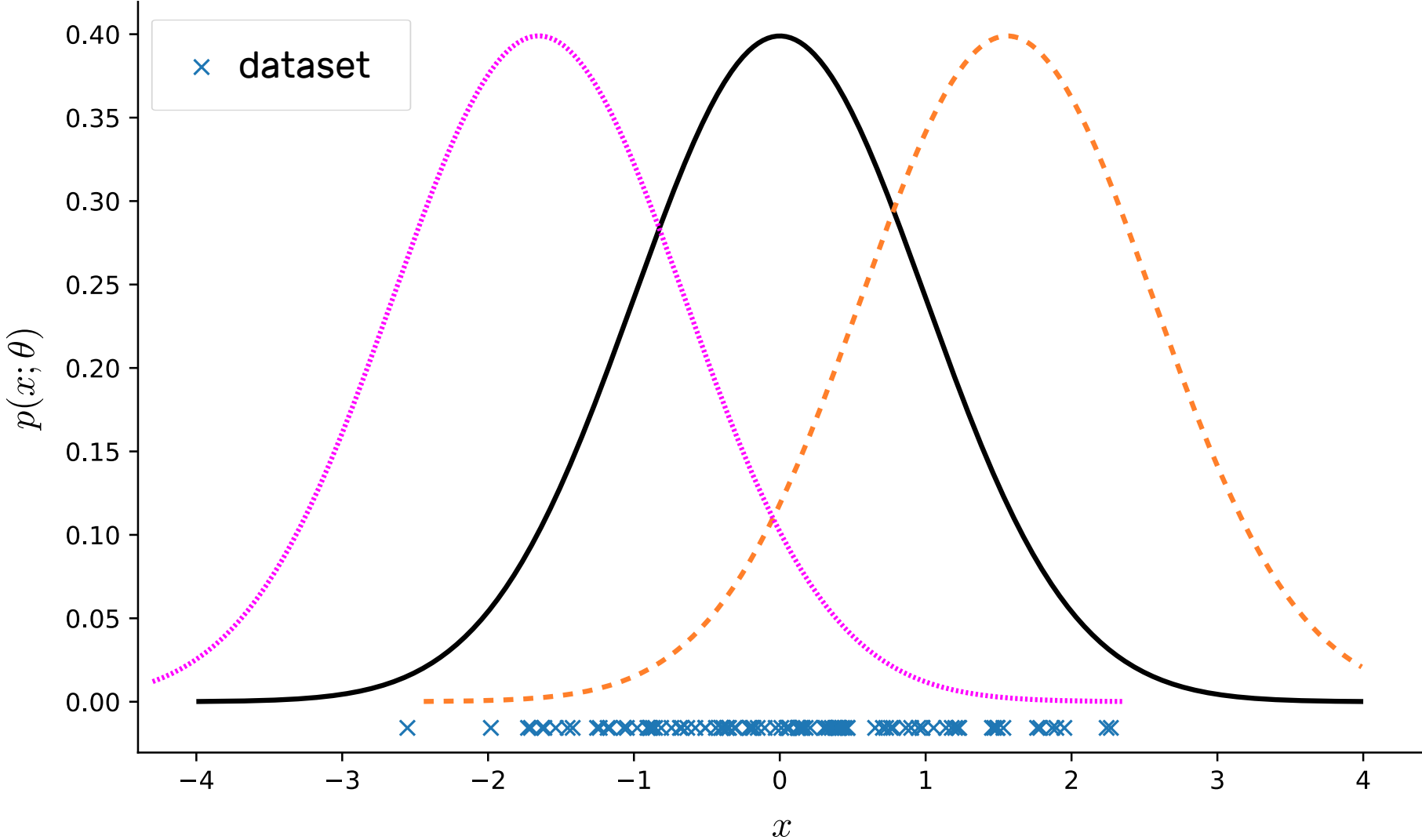
- We use the **Monte Carlo method**, which approximates the intractable expectation by an empirical average using i.i.d samples drawn from $p^*(\mathbf{x})$:

$$\mathbb{E}_{p^*(\mathbf{x})} [\ln p(\mathbf{x}; \theta)] \approx \frac{1}{N} \sum_{i=1}^N \ln p(\mathbf{x}_i; \theta).$$

- This last expression shows that **choosing the Kullback-Leibler divergence as the measure of fit leads to maximum (log-marginal) likelihood parameters estimation.**

We are trying to find the model parameters that are the most likely on average over the dataset, where "being likely" means that the corresponding log-density $\ln p(\mathbf{x}; \theta)$ is high when evaluated on the samples of the dataset.

Which distribution better fits the data?



Summary

- **Data:** Get the dataset $\mathcal{D} = \left\{ \mathbf{x}_i \stackrel{i.i.d}{\sim} p^*(\mathbf{x}) \right\}_{i=1}^N$.
- **Modeling:** Define a model that relates the latent variable of interest to the observations $p(\mathbf{x}, z; \theta) = p(\mathbf{x} | z; \theta_x)p(z; \theta_z)$.
- **Inference:** Compute the posterior distribution $p(z | \mathbf{x}; \theta)$, which can then be used in many different ways.
- **Learning:** Estimate the unknown model parameters θ by maximizing the log-marginal likelihood $\ln p(\mathbf{x}; \theta)$ averaged over the dataset.

Taking a step back and looking at the landscape of machine learning

What we have seen so far actually corresponds to a subset of machine learning methods, involving

- **generative modeling**, because we define a generative model of the observed data;
- **Bayesian modeling and inference**, because the generative model involves a latent random variable equipped with a prior and during inference we compute its posterior distribution;
- **unsupervised learning**, because the parameters of the model, which *in fine* allow us to infer the latent variable of interest from the observations through the posterior, are learned from unlabeled data.

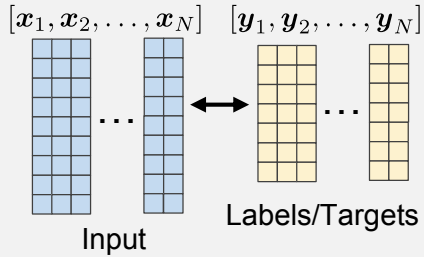
Supervised learning is another important subset of machine learning methods, which involves generative or **discriminative models**. This will be the topic of another lecture.

- Supervised learning with discriminative models is probably the dominating paradigm in machine learning, which has led to great research and industrial successes in recent years.
- But understanding first unsupervised learning with generative models greatly helps to have a deep understanding of supervised learning with discriminative models.
- This is for three reasons:
 1. The whole story of extracting a latent variable of interest from observations is always valid **at test time**, whatever the machine learning method.
 2. Supervised learning is simply the case where, **at training time**, the variable of interest is not latent anymore but observed and used for the learning of the model parameters (no need to marginalize it anymore!);
 3. Discriminative modeling is simply the case where we directly define the posterior distribution in the modeling step, instead of defining the joint distribution and then using Bayes theorem.

Supervised Learning

Labeled training data

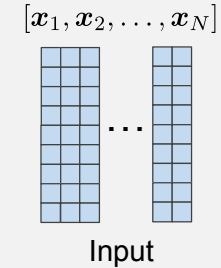
$$\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$$



Unsupervised Learning

Unlabeled training data

$$\{\mathbf{x}_i\}_{i=1}^N$$



Unlabeled test data \mathbf{x}

Learned model

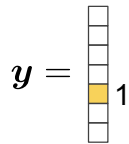
θ

\mathbf{y} or \mathbf{z}

1. Discrete case: («one-hot»)

► Classification

Ex. application: *dog breed*



13: German Shepherd

2. Continuous case:

► Regression

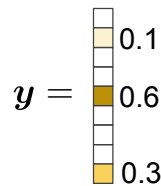
Ex. application: *head pose*



3. Sparse case:

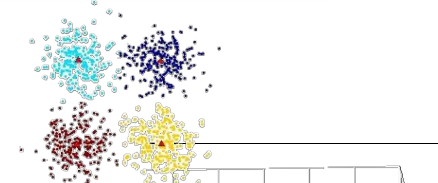
► Multi-classification

Ex. application: *image labelling*



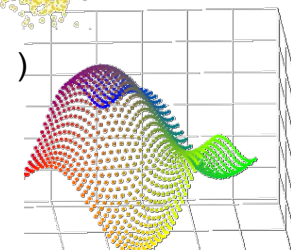
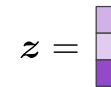
1. Discrete case:

► Clustering



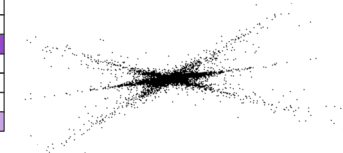
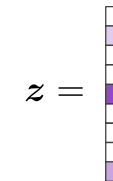
2. Continuous case: ($\dim(\mathbf{z}) \ll \dim(\mathbf{x})$)

► Dimensionality Reduction

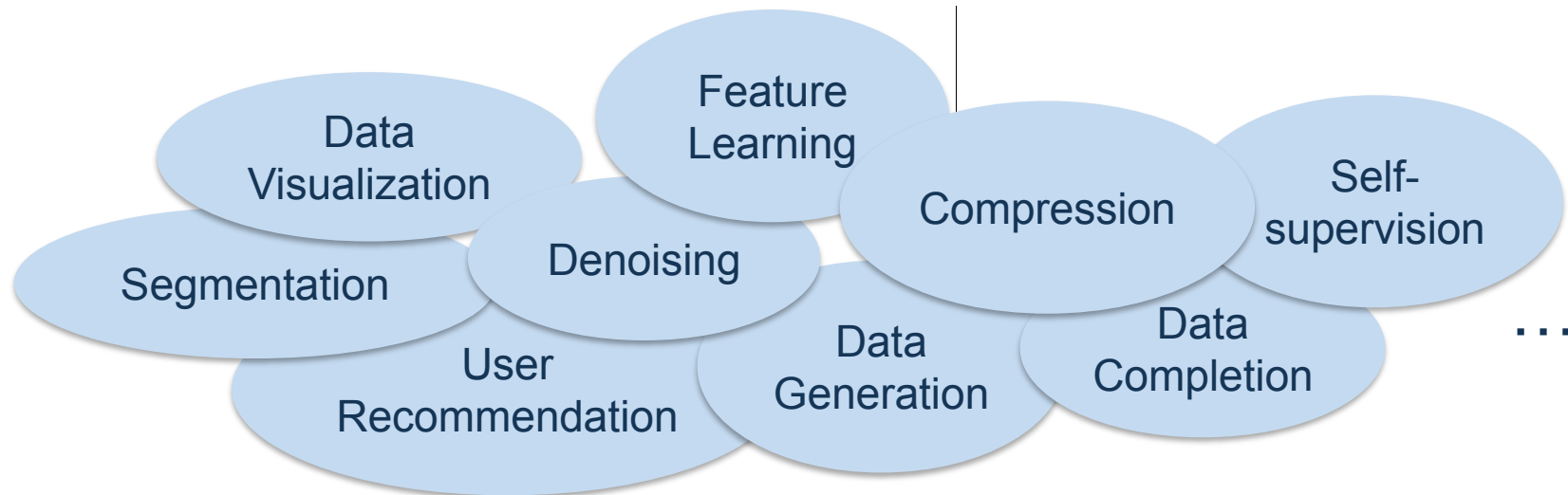


3. Sparse case:

► Dictionary Learning

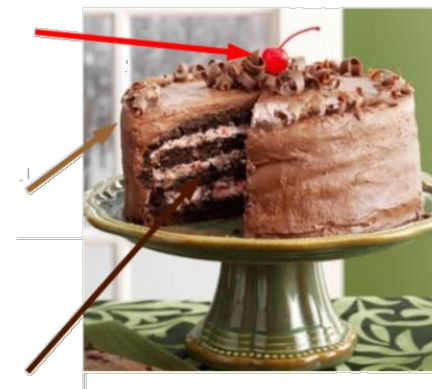


Applications of unsupervised learning






« If AI was a cake, reinforcement learning would be the cherry on the cake, supervised learning the icing, and unsupervised learning the génoise. »

-Yann Lecun (Facebook AI) at NIPS 2016



Potential to learn from massive amount of unlabeled data to generate even more.

Fundamental techniques in unsupervised learning

	Clustering	Dim. Reduction	Dict. Learning
z	Discrete 	Continuous 	Sparse 
Techniques	<ul style="list-style-type: none"> • K-means • GMM EM 	<ul style="list-style-type: none"> • PCA • Manifold Learning 	<ul style="list-style-type: none"> • Sparse Coding • K-SVD

These fundamental techniques can all be described from a probabilistic perspective, where

- the structure of the latent variable of interest z is encoded in a suitable probabilistic prior (**modeling**);
- the task of extracting z from the observations corresponds to the computation of its posterior (**inference**);
- the model parameters are estimated by maximizing the marginal likelihood (**learning**).

Machine learning for signal processing

When the observations or the latent variables correspond to natural signals or images, we are actually doing signal processing using machine learning.

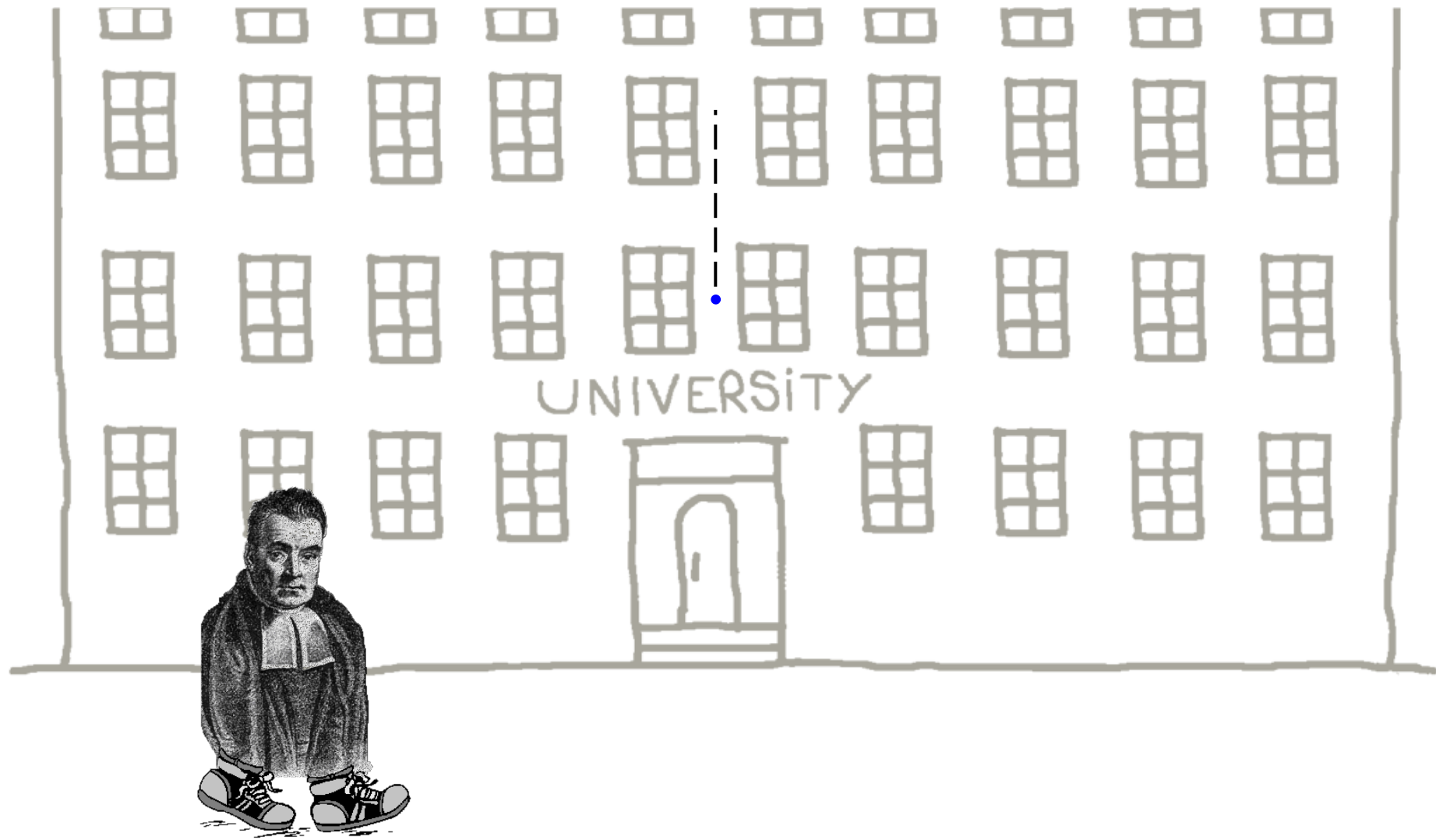
The previous fundamental techniques form the basis of many advanced deep learning techniques used today:

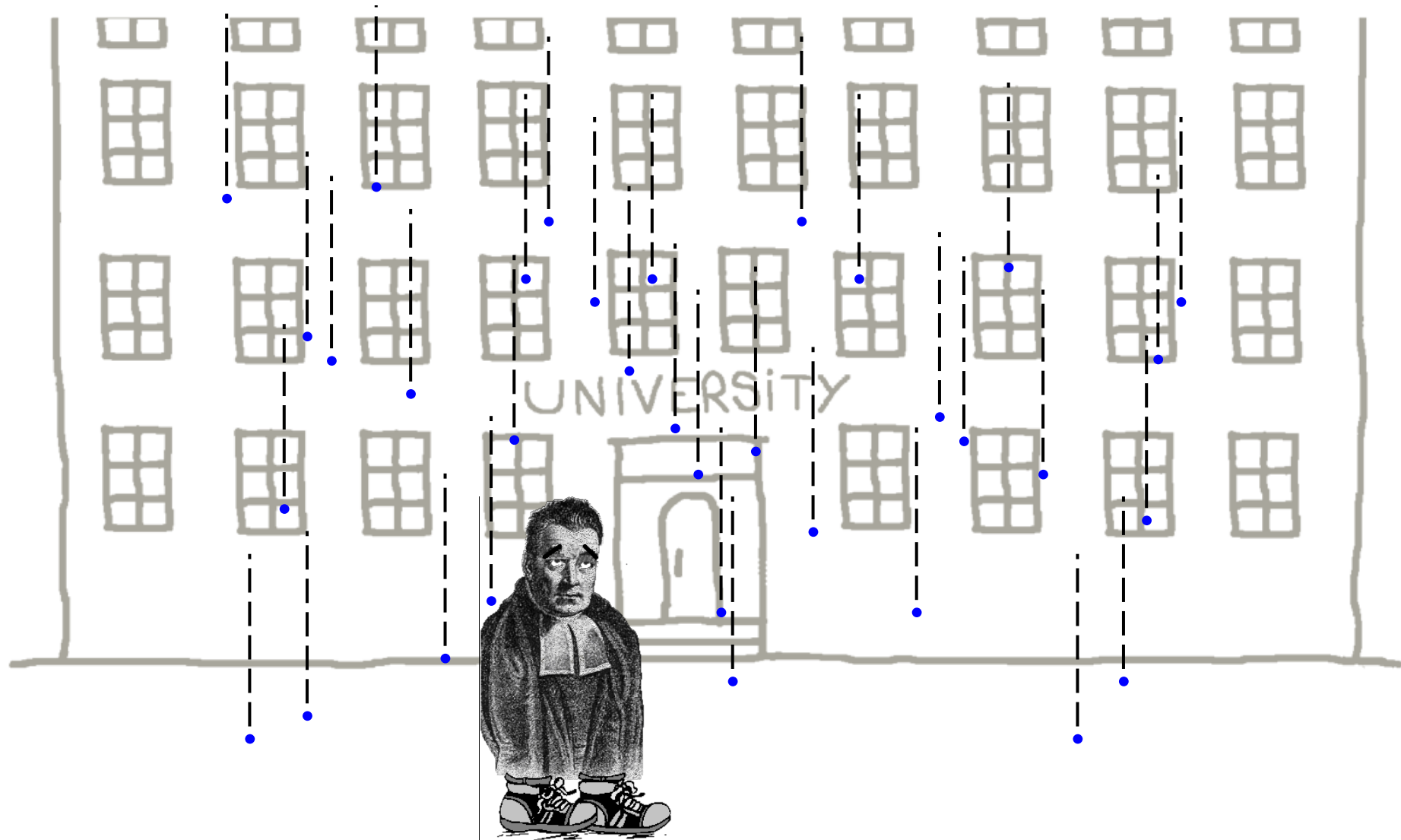
- Variational autoencoders (VAEs);
- Generative adversarial networks (GANs);
- Normalizing flow;
- Diffusion models;
- Conditional neural processes;
- Self-supervised learning;
- etc.

In the adventures of Thomas Bayes, episode 1, we discovered the modeling and inference steps, but not the learning step (the model parameters were assumed to be known and fixed).

The adventures of Thomas Bayes, episode 2

The following example and drawings are adapted from a [tutorial on Bayesian Learning for Signal Processing](#) given by Antoine Deleforge at the LVA/ICA 2015 Summer School.





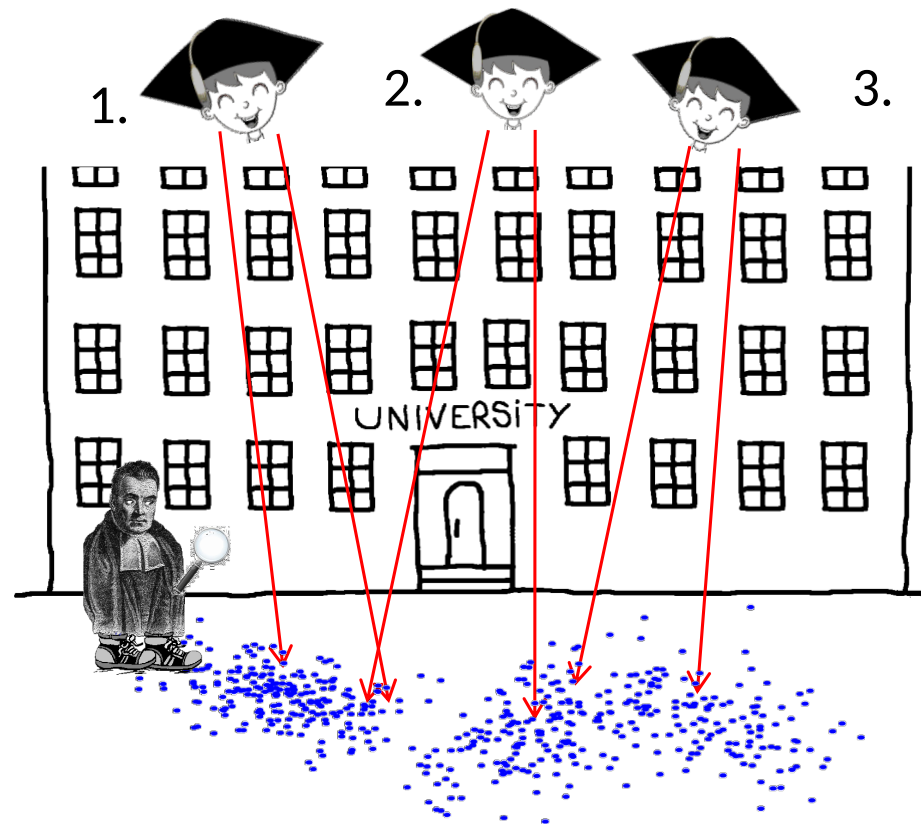
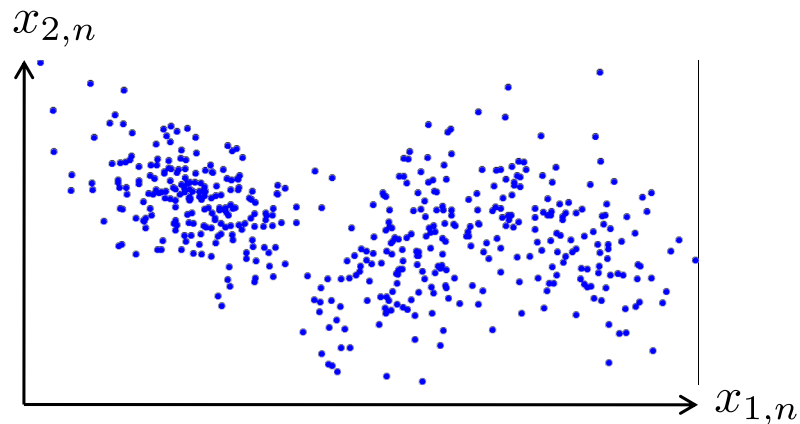




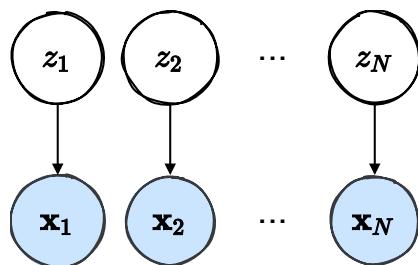
Modeling

Observed variables: $\{\mathbf{x}_i \in \mathbb{R}^2\}_{i=1}^N$.

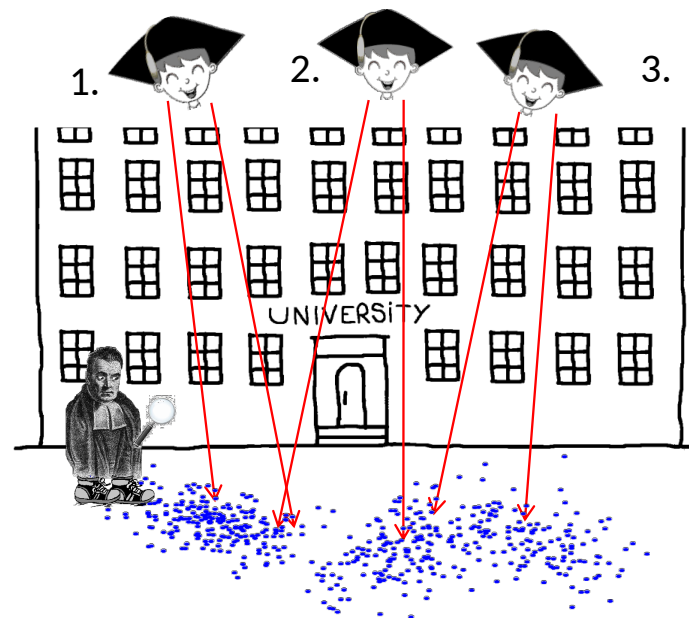
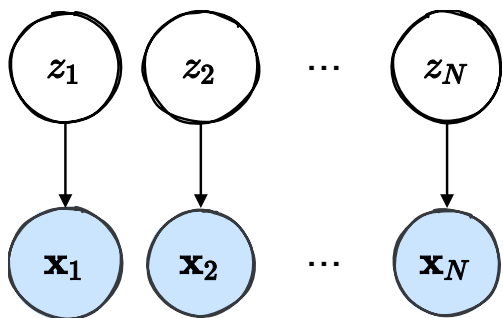
Hidden variables: $\{z_i \in \{1, 2, 3\}\}_{i=1}^N$.



Generative model



$$p(\{\mathbf{x}_i, z_i\}_{i=1}^N; \theta) = \prod_{i=1}^N p(\mathbf{x}_i, z_i; \theta) = \prod_{i=1}^N p(\mathbf{x}_i | z_i; \theta) p(z_i; \theta).$$



Prior

$$p(z_i = k; \theta) = \pi_k, \quad \sum_{k=1}^K \pi_k = 1, \quad K = 3$$

Likelihood

$$p(\mathbf{x}_i | z_i = k; \theta) = \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Parameters

$$\theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K.$$

Marginal likelihood

$$\begin{aligned} p(\{\mathbf{x}_i\}_{i=1}^N; \theta) &= \prod_{i=1}^N p(\mathbf{x}_i; \theta) \\ &= \prod_{i=1}^N \sum_{k=1}^K p(\mathbf{x}_i \mid z_i = k; \theta) p(z_i = k; \theta) \\ &= \prod_{i=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \end{aligned}$$

Observations are independent and identically distributed according to a **Gaussian mixture model** (GMM) with $K = 3$ components.

The parameters π_k are called the **mixing coefficients**, they give the prior probability of picking the k -th component to generate a data point \mathbf{x}_i .

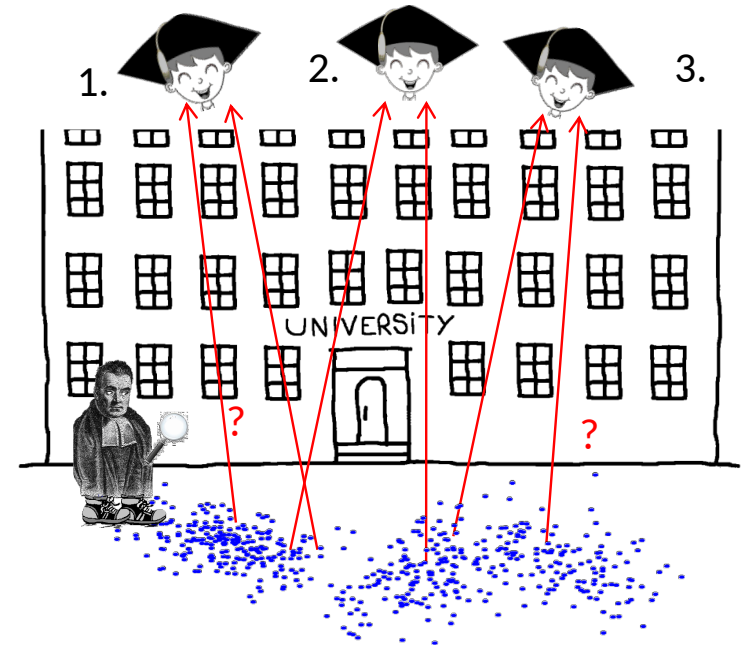
Inference

Posterior distribution

$$p(\{z_i\}_{i=1}^N | \{\mathbf{x}_i\}_{i=1}^N; \theta) = \prod_{i=1}^N p(z_i | \mathbf{x}_i; \theta),$$

where

$$\begin{aligned} p(z_i = k | \mathbf{x}_i; \theta) &= \frac{p(\mathbf{x}_i | z_i = k; \theta)p(z_i = k; \theta)}{p(\mathbf{x}_i; \theta)} \\ &= \frac{p(\mathbf{x}_i | z_i = k; \theta)p(z_i = k; \theta)}{\sum_{j=1}^K p(\mathbf{x}_i | z_i = j; \theta)p(z_i = j; \theta)} \\ &= \frac{\pi_k p(\mathbf{x}_i | z_i = k; \theta)}{\sum_{j=1}^K \pi_j p(\mathbf{x}_i | z_i = j; \theta)}. \end{aligned}$$



The posterior probabilities $p(z_i = k | \mathbf{x}_i; \theta)$ are also known as the **responsibilities**.

The argmax of the responsibility assigns the observation to a component, i.e. it **clusters the data**.

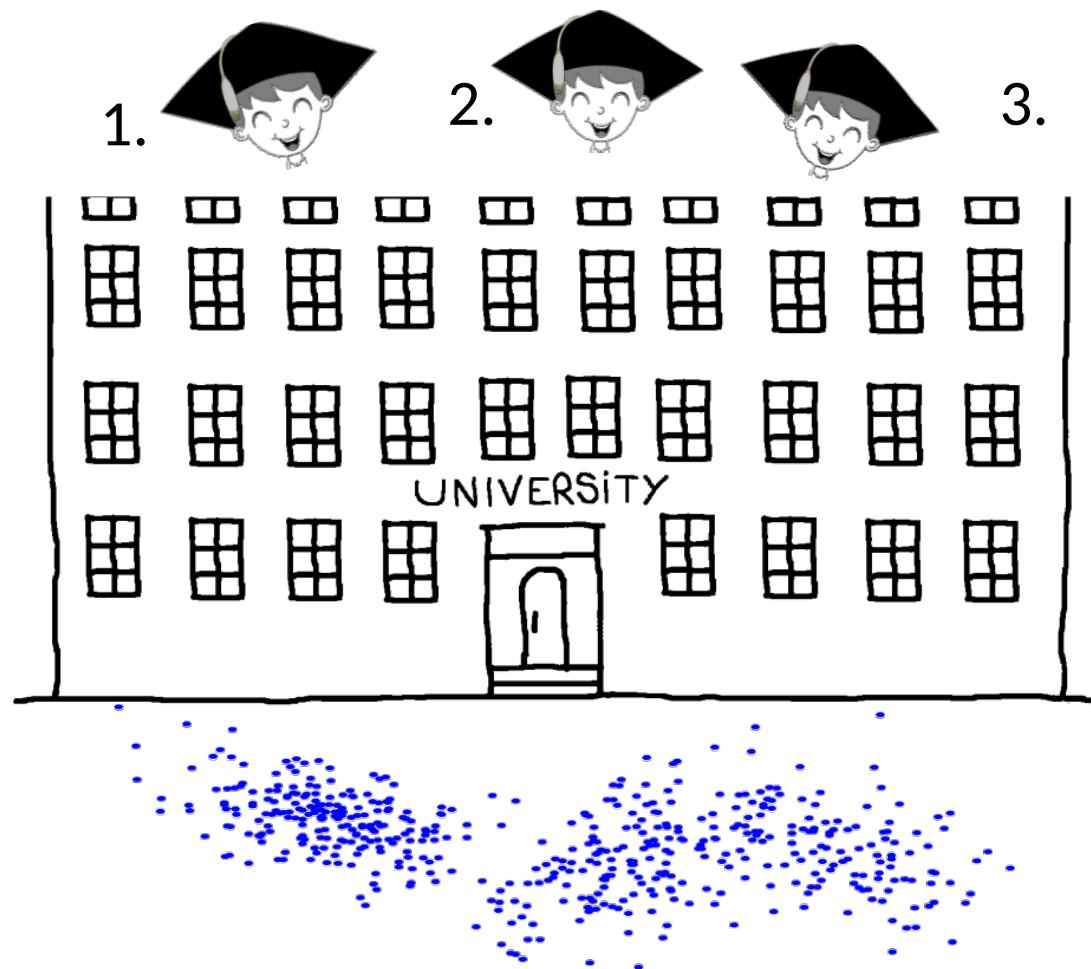
Parameters estimation

The posterior distribution can be computed analytically, but it depends on the **unknown model parameters** $\theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$.

Ideally, we would like to estimate them by maximizing the log-marginal likelihood:

$$\begin{aligned}\mathcal{L}(\theta) &= \ln p(\{\mathbf{x}_i\}_{i=1}^N; \theta) \\ &= \ln \prod_{i=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \sum_{i=1}^N \ln \left(\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right).\end{aligned}$$

Due to the presence of the sum over k inside the logarithm, **the maximum marginal likelihood solution for the parameters does not have a closed-form analytical solution.**



Let's derive an EM algorithm

The expectation-maximization algorithm

The expectation-maximization (EM) algorithm is a general technique introduced by Dempster et al. in 1977 for maximum likelihood parameters estimation in probabilistic models having latent variables.

Let $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$ denote the **observed and latent** random variables, respectively, which are assumed to be continuous, although the discussion is identical in the discrete setting.

We assume that direct optimization of the marginal likelihood $p(\mathbf{x}; \theta)$ is difficult, while optimization of the complete-data likelihood function $p(\mathbf{x}, \mathbf{z}; \theta)$ is much simpler.

The evidence lower bound

We first introduce a distribution over the latent variables whose probability density function is denoted by $q(\mathbf{z})$.

For any distribution $q(\mathbf{z})$, the following decomposition of the log-marginal likelihood holds:

$$\ln p(\mathbf{x}; \theta) = \mathcal{L}(q(\mathbf{z}), \theta) + D_{\text{KL}}(q(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x}; \theta)),$$

where $\mathcal{L}(q(\mathbf{z}), \theta)$ is called the **evidence lower bound** (ELBO), and it is defined by

$$\mathcal{L}(q(\mathbf{z}), \theta) = \mathbb{E}_{q(\mathbf{z})}[\ln p(\mathbf{x}, \mathbf{z}; \theta) - \ln q(\mathbf{z})].$$

The **Kullback-Leibler** (KL) divergence is defined by:

$$D_{\text{KL}}(q \parallel p) = \mathbb{E}_q[\ln(q) - \ln(p)],$$

and it satisfies $D_{\text{KL}}(q \parallel p) \geq 0$ with equality if and only if $q = p$.

Proof: $\ln p(\mathbf{x}; \theta) = \mathbb{E}_{q(\mathbf{z})}[\ln p(\mathbf{x}; \theta)] = \mathbb{E}_{q(\mathbf{z})}[\ln p(\mathbf{x}, \mathbf{z}; \theta) - \ln p(\mathbf{z} \mid \mathbf{x}; \theta) - \ln q(\mathbf{z}) + \ln q(\mathbf{z})]$

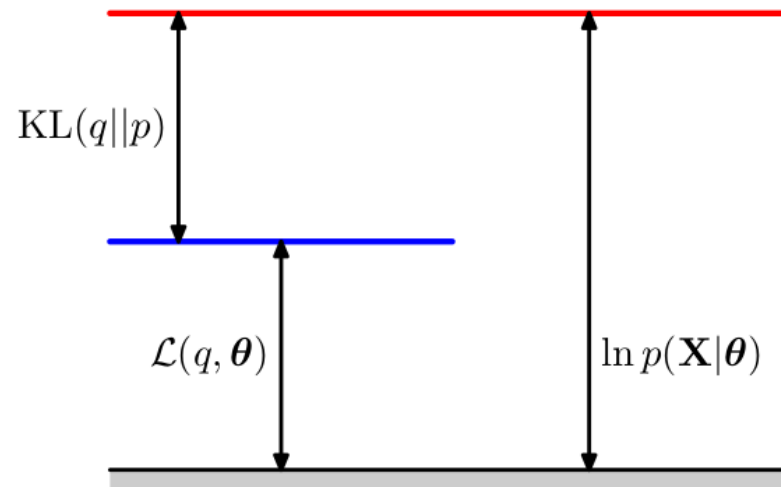
$$\ln p(\mathbf{x}; \theta) = \mathcal{L}(q(\mathbf{z}), \theta) + D_{\text{KL}}(q(\mathbf{z}) \parallel p(\mathbf{z} | \mathbf{x}; \theta))$$

As the KL divergence is always non-negative, we have:

$$\ln p(\mathbf{x}; \theta) \geq \mathcal{L}(q(\mathbf{z}), \theta),$$

with equality if and only if $q(\mathbf{z}) = p(\mathbf{z} | \mathbf{x}; \theta)$.

The ELBO is indeed a **lower bound of the log-marginal likelihood**, which is tight if $q(\mathbf{z})$ matches the true posterior.



EM algorithm

The EM algorithm is an iterative algorithm which alternates between optimizing the ELBO with respect to q in the E-Step and with respect to θ in the M-step.

We first initialize θ_0 , then we iterate for $t \geq 0$

- **E-Step:** $q_{t+1}(\mathbf{z}) = \arg \max_q \mathcal{L}(q(\mathbf{z}), \theta_t)$
- **M-Step:** $\theta_{t+1} = \arg \max_{\theta} \mathcal{L}(q_{t+1}(\mathbf{z}), \theta)$

E-Step

We recall the decomposition of the log-marginal likelihood:

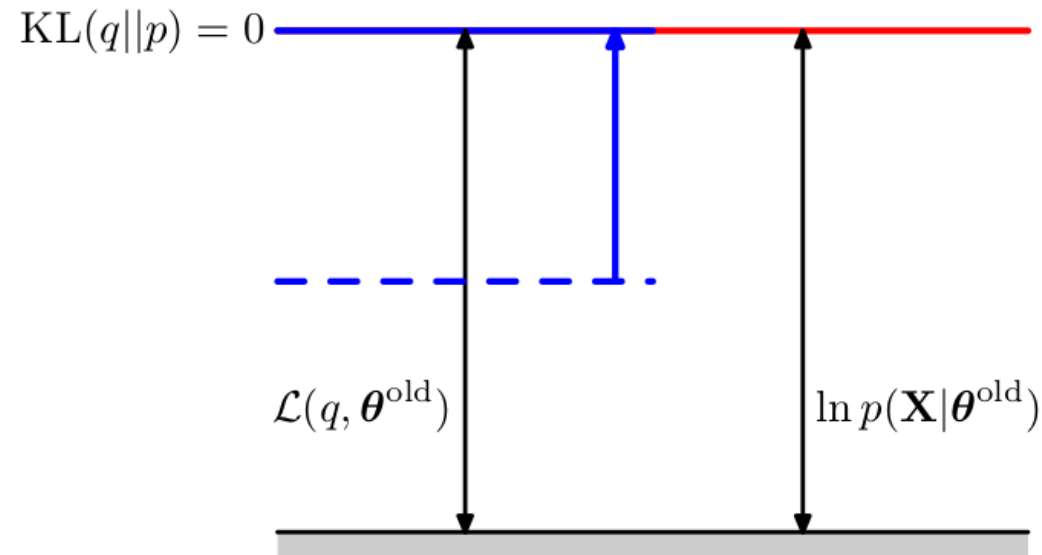
$$\ln p(\mathbf{x}; \theta) = \mathcal{L}(q(\mathbf{z}), \theta) + D_{\text{KL}}(q(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x}; \theta)).$$

The solution of the E-step is given by:

$$\begin{aligned} q_{t+1}(\mathbf{z}) &= \arg \max_q \mathcal{L}(q(\mathbf{z}), \theta_t) \\ &= \arg \min_q D_{\text{KL}}(q(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x}; \theta_t)) \\ &= p(\mathbf{z} \mid \mathbf{x}; \theta_t). \end{aligned}$$

After the E-Step, we have $D_{\text{KL}}(q_{t+1}(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x}; \theta_t)) = 0$, and the ELBO is equal to the log-marginal likelihood (i.e. the lower-bound is tight):

$$\ln p(\mathbf{x}; \theta_t) = \mathcal{L}(q_{t+1}(\mathbf{z}), \theta_t).$$



Therefore, maximizing the lower-bound with respect to the model parameters in the M-step will necessarily increase the log-marginal likelihood.

M-Step

- We recall the expression of the ELBO:

$$\mathcal{L}(q(\mathbf{z}), \theta) = \mathbb{E}_{q(\mathbf{z})}[\ln p(\mathbf{x}, \mathbf{z}; \theta) - \ln q(\mathbf{z})],$$

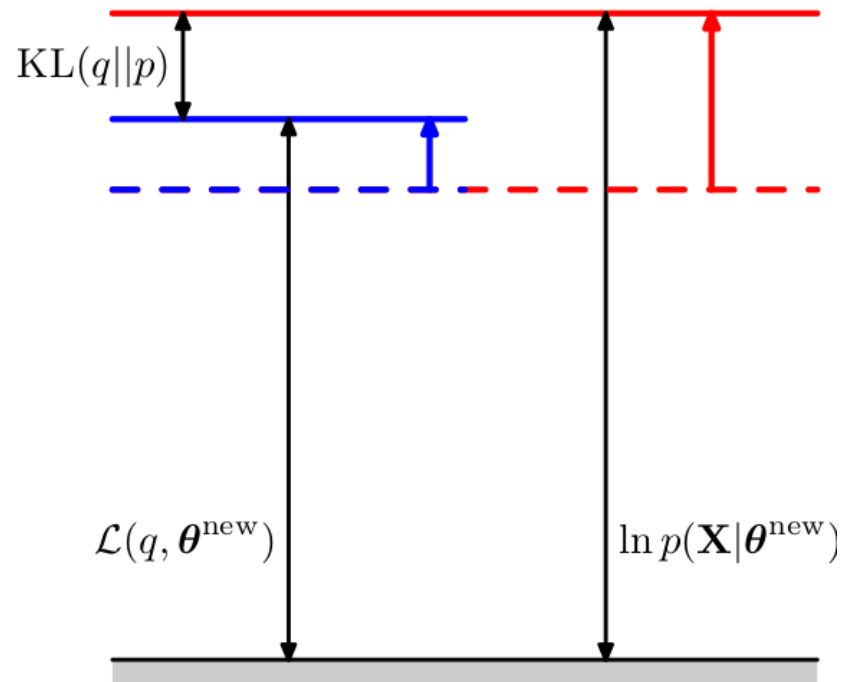
- The solution of the M-step is given by:

$$\begin{aligned}\theta_{t+1} &= \arg \max_{\theta} \mathcal{L}(q_{t+1}(\mathbf{z}), \theta) \\ &= \arg \max_{\theta} \mathcal{L}(p(\mathbf{z} | \mathbf{x}; \theta_t), \theta) \\ &= \arg \max_{\theta} \mathbb{E}_{p(\mathbf{z} | \mathbf{x}; \theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta) - \ln p(\mathbf{z} | \mathbf{x}; \theta_t)] \\ &= \arg \max_{\theta} \mathbb{E}_{p(\mathbf{z} | \mathbf{x}; \theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)] + cst(\theta),\end{aligned}$$

where the constant is the differential entropy of $p(\mathbf{z} | \mathbf{x}; \theta_t)$ which is independent of θ .

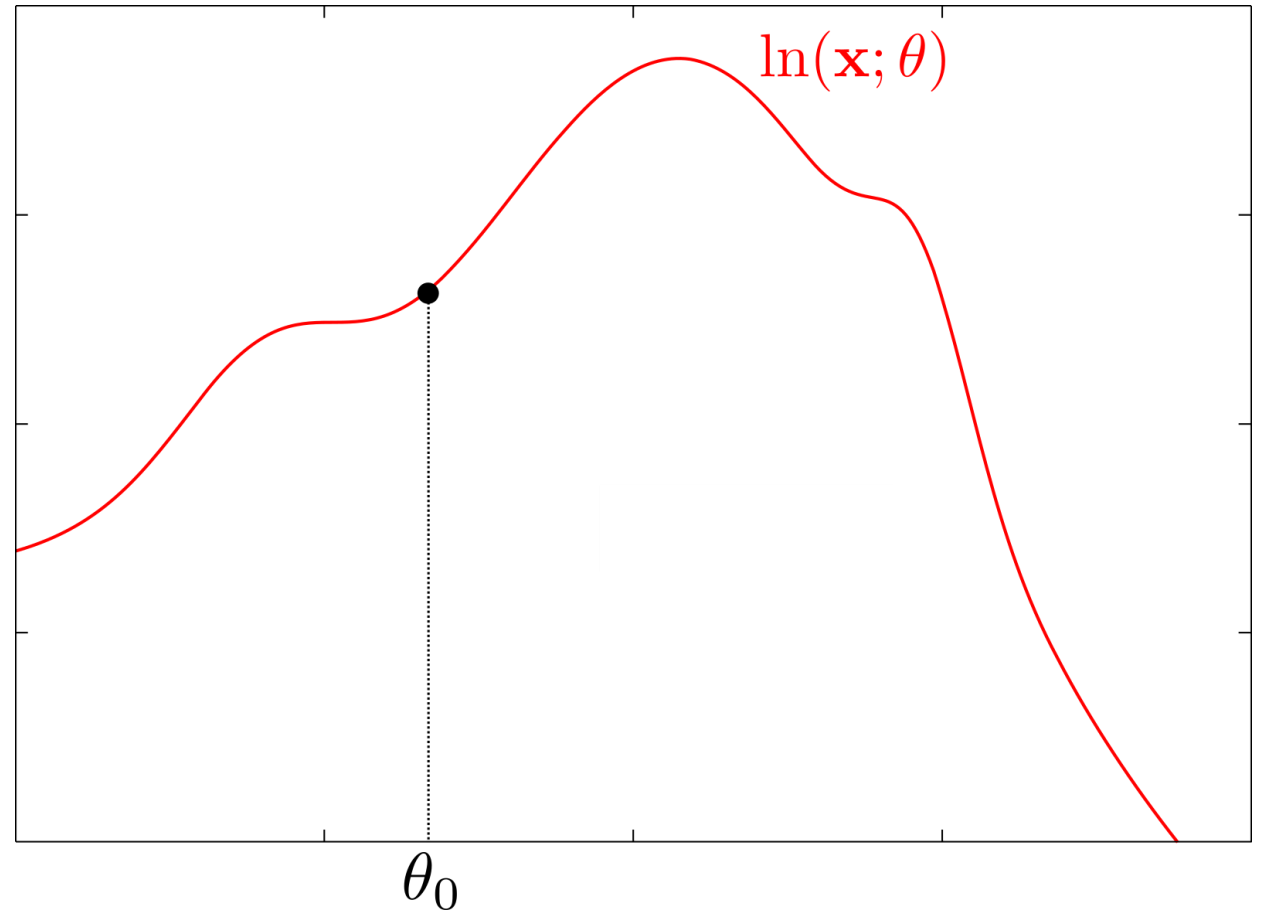
After the M-step, because $q_{t+1}(\mathbf{z}) = p(\mathbf{z} | \mathbf{x}; \theta_t)$ has been held fixed for computing the new model parameters θ_{t+1} , the KL divergence $D_{\text{KL}}(q_{t+1}(\mathbf{z}) || p(\mathbf{z} | \mathbf{x}; \theta_{t+1}))$ will be non zero.

The increase in the log-marginal likelihood function is therefore greater than the increase in the ELBO, as shown below.



We recall the decomposition of the log-marginal likelihood $\ln p(\mathbf{x}; \theta) = \mathcal{L}(q(\mathbf{z}), \theta) + D_{\text{KL}}(q(\mathbf{z}) || p(\mathbf{z} | \mathbf{x}; \theta))$.

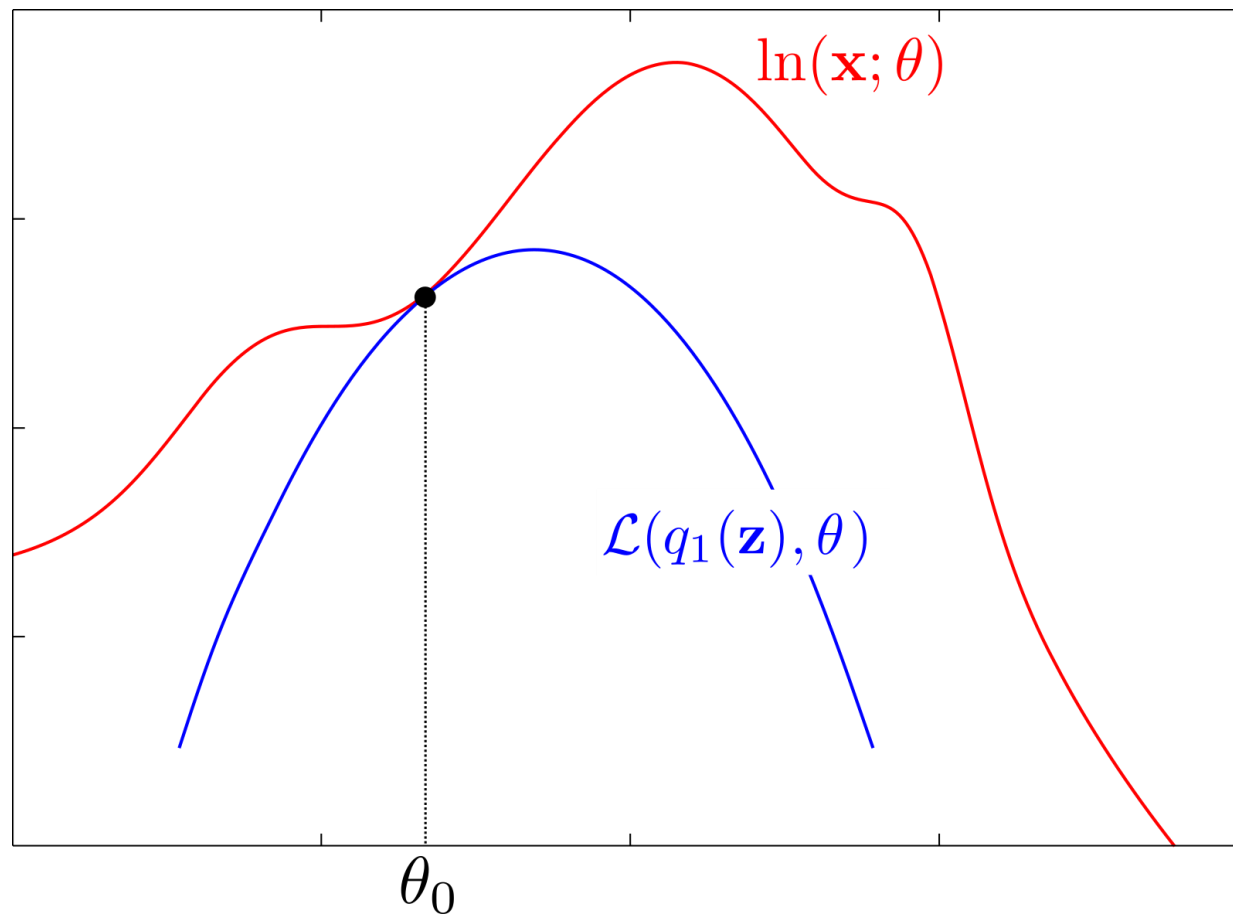
Initialize θ_0 .



Iteration $t = 1$:

- E-Step:

$$q_1(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}; \theta_0)$$



We have $D_{\text{KL}}(q_1(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x}; \theta_0)) = 0$ such that $\ln p(\mathbf{x}; \theta_0) = \mathcal{L}(q_1(\mathbf{z}), \theta_0)$.

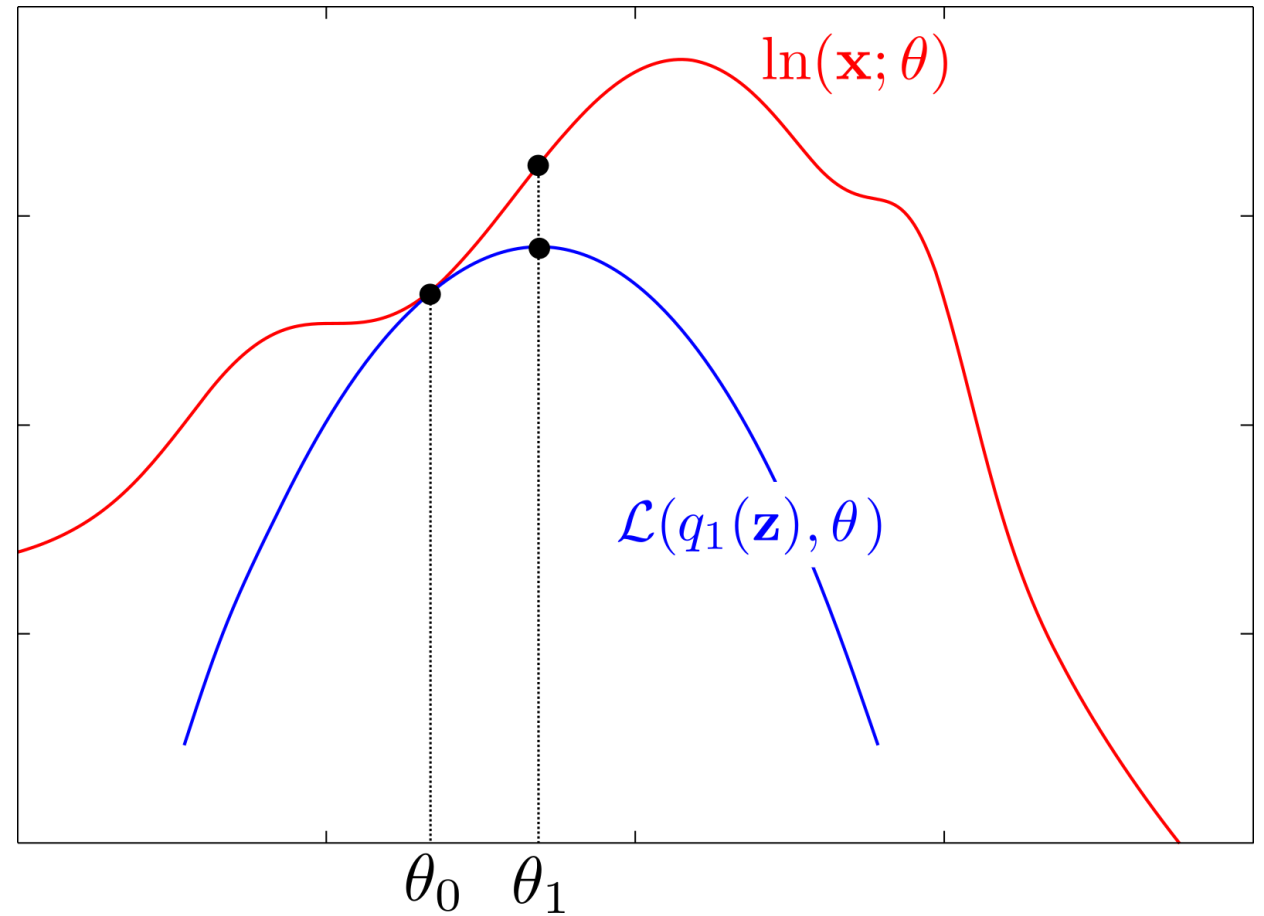
Iteration $t = 1$:

- E-Step:

$$q_1(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}; \theta_0)$$

- M-Step:

$$\theta_1 = \arg \max_{\theta} \mathcal{L}(q_1(\mathbf{z}), \theta)$$



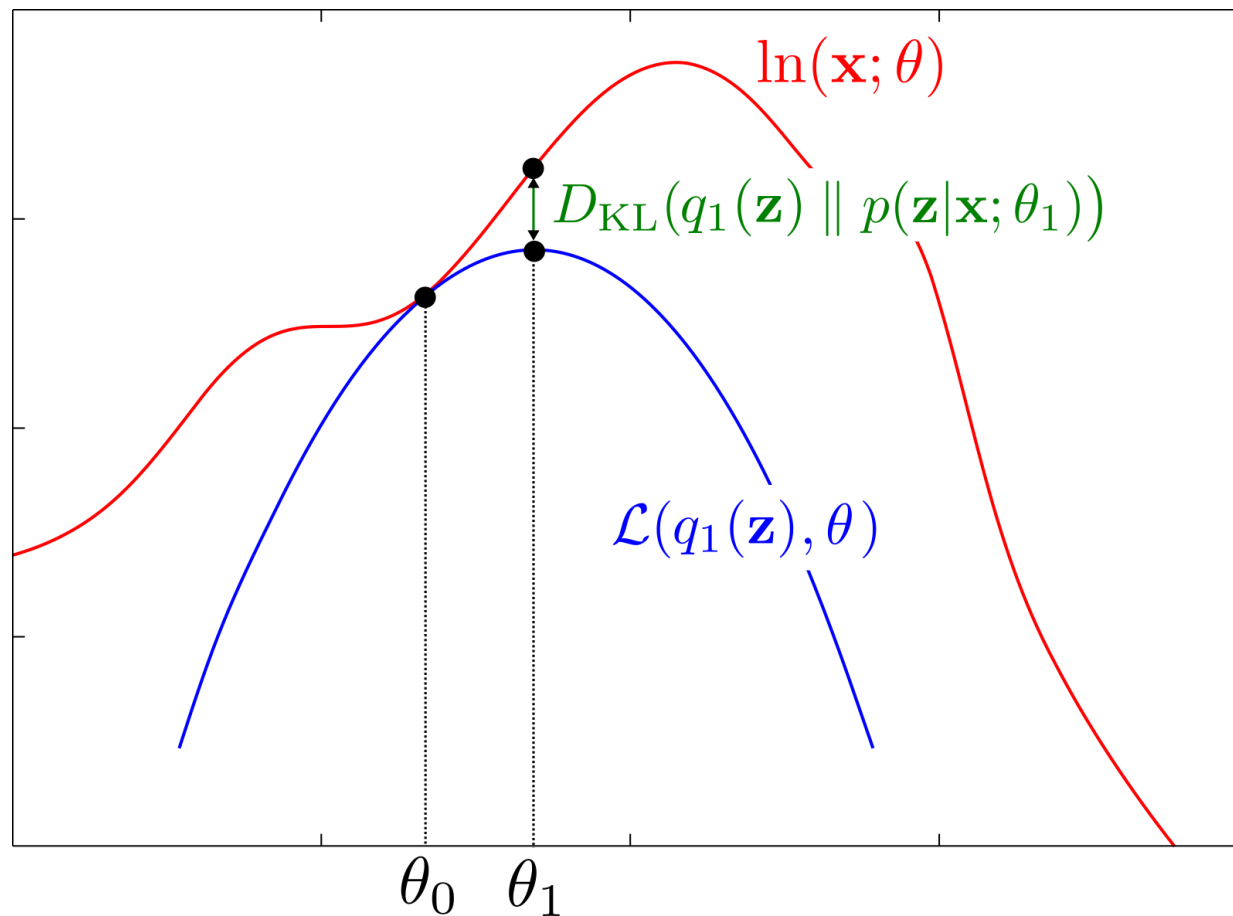
Iteration $t = 1$:

- E-Step:

$$q_1(\mathbf{z}) = p(\mathbf{z} | \mathbf{x}; \theta_0)$$

- M-Step:

$$\theta_1 = \arg \max_{\theta} \mathcal{L}(q_1(\mathbf{z}), \theta)$$

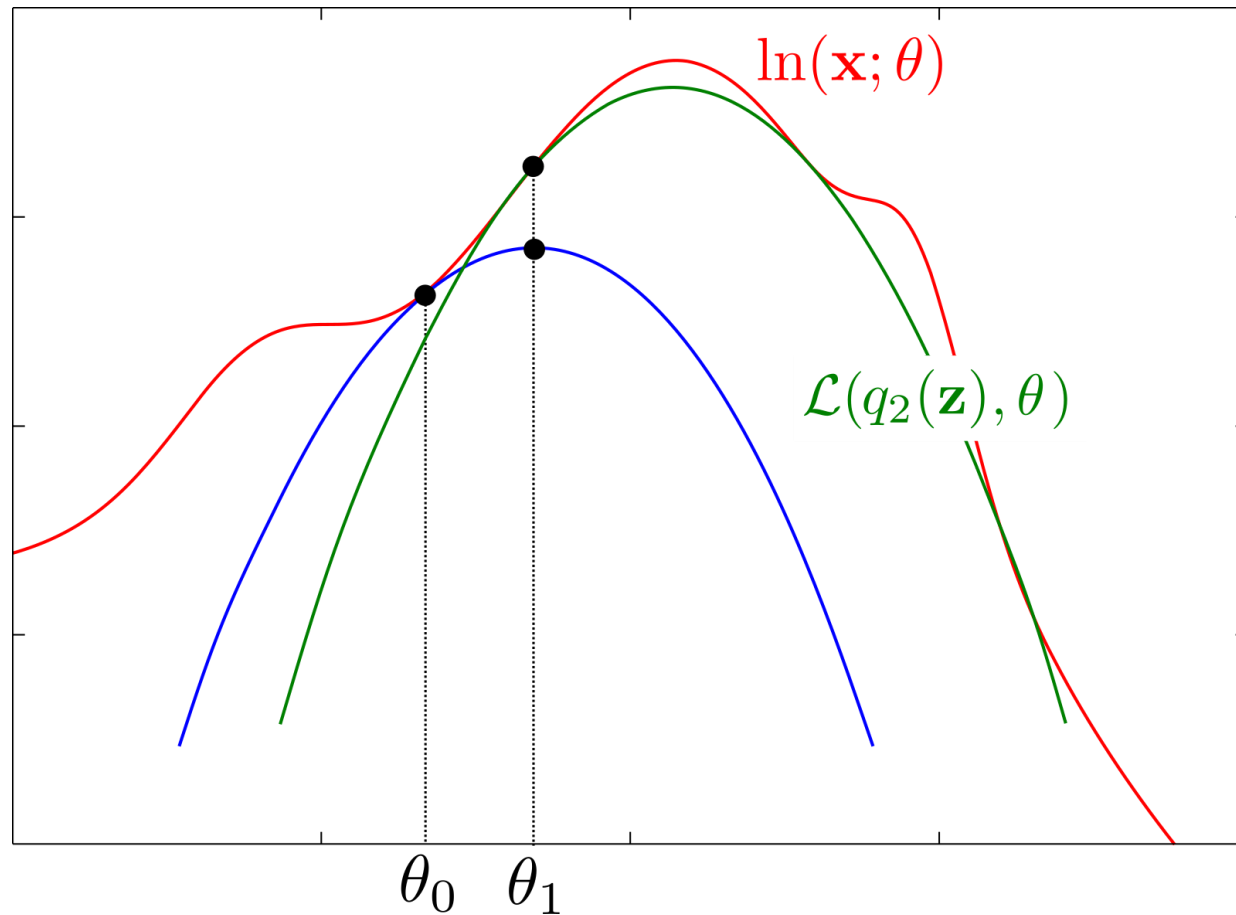


We have $D_{\text{KL}}(q_1(\mathbf{z}) \parallel p(\mathbf{z} | \mathbf{x}; \theta_1)) \neq 0$.

Iteration $t = 2$:

- E-Step:

$$q_2(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}; \theta_1)$$



We have $D_{\text{KL}}(q_2(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x}; \theta_1)) = 0$ such that $\ln p(\mathbf{x}; \theta_1) = \mathcal{L}(q_2(\mathbf{z}), \theta_1)$.

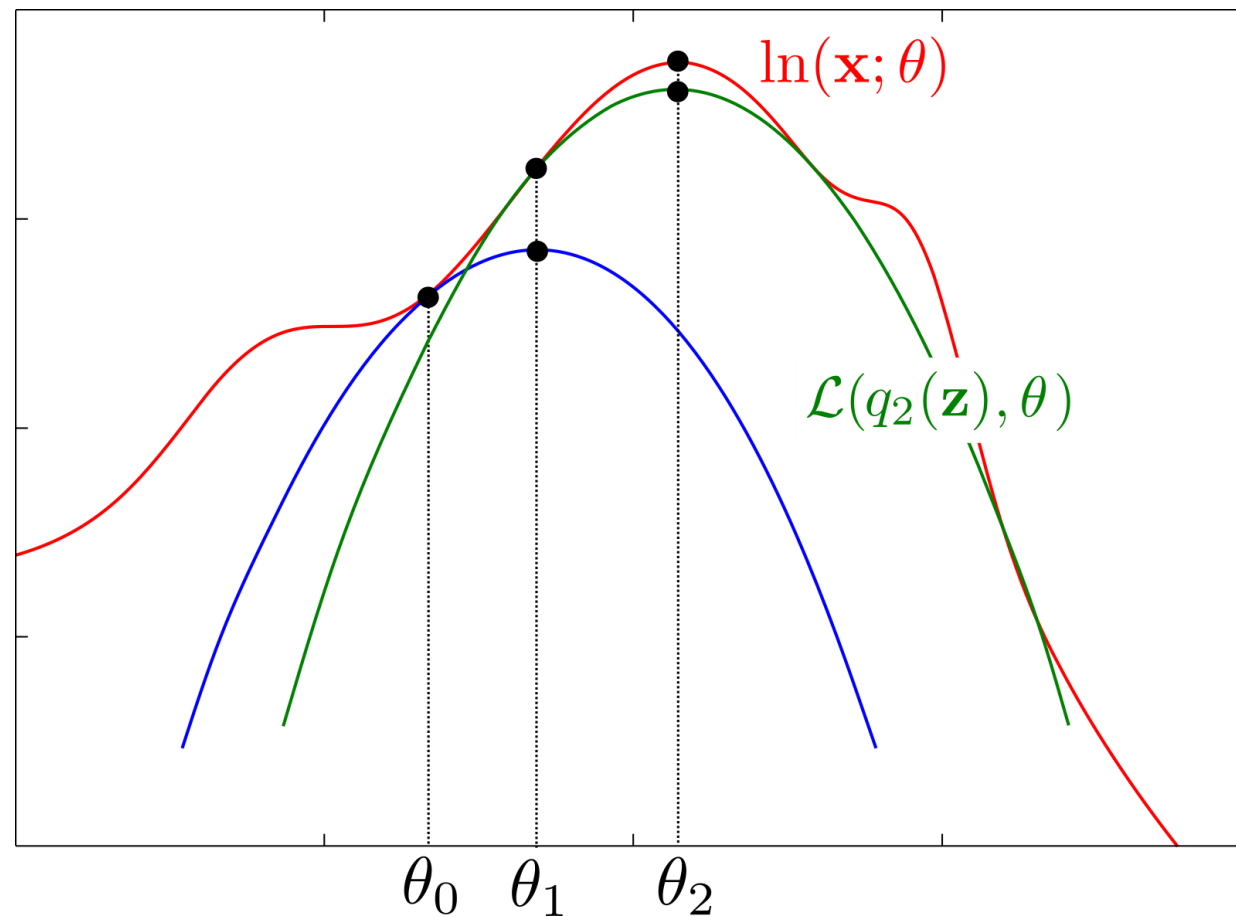
Iteration $t = 2$:

- E-Step:

$$q_2(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}; \theta_1)$$

- M-Step:

$$\theta_2 = \arg \max_{\theta} \mathcal{L}(q_2(\mathbf{z}), \theta)$$



Iteration $t = 2$:

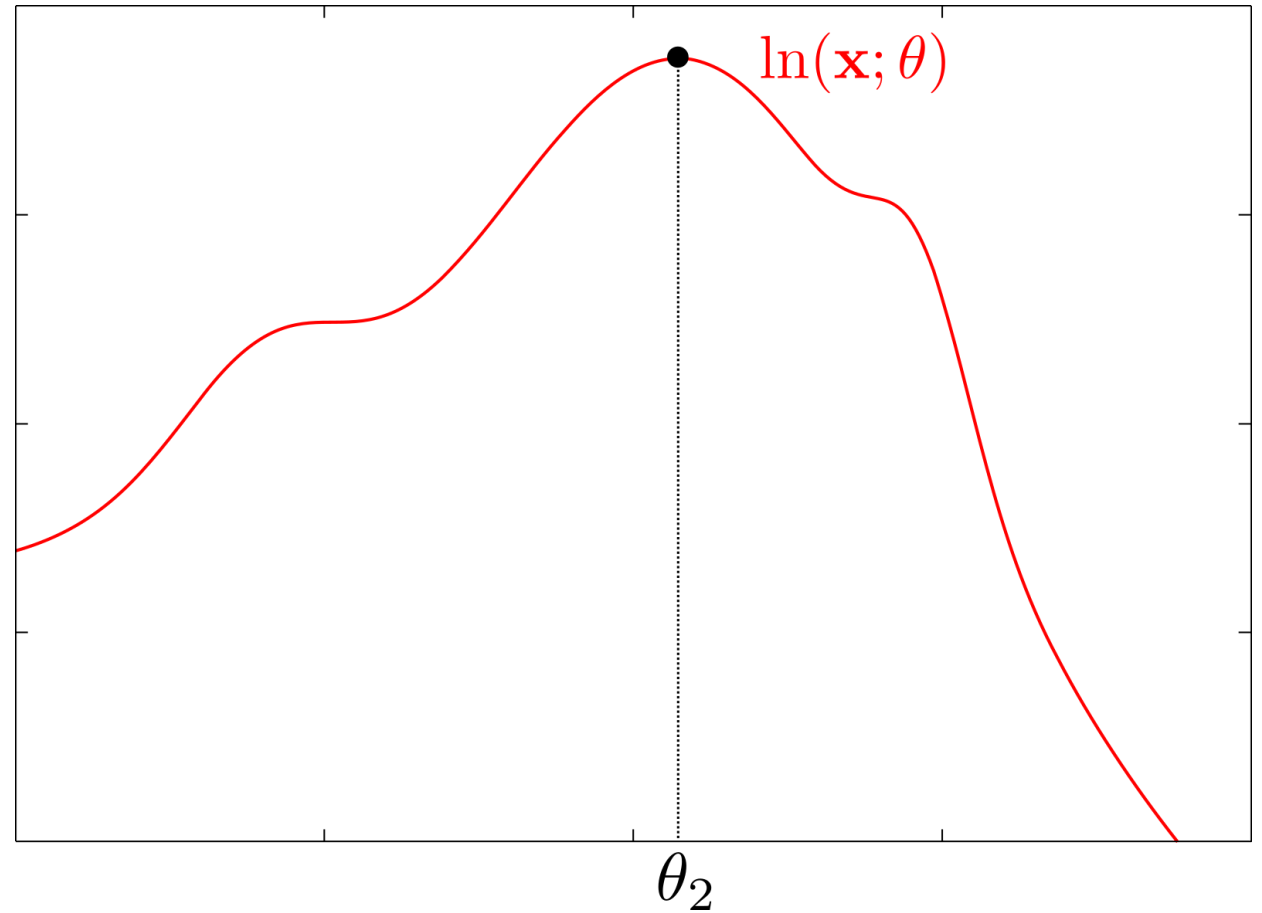
- E-Step:

$$q_2(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}; \theta_1)$$

- M-Step:

$$\theta_2 = \arg \max_{\theta} \mathcal{L}(q_2(\mathbf{z}), \theta)$$

- We reached a stationary point.



Properties of the EM algorithm

- The log-marginal likelihood is **monotonically increasing**.

Properties of the EM algorithm

- The log-marginal likelihood is **monotonically increasing**.
-

Proof:

Using the fact that $\ln p(\mathbf{x}; \theta) = \mathcal{L}(q(\mathbf{z}), \theta) + D_{\text{KL}}(q(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x}; \theta))$ and $q_{t+1}(\mathbf{z}) = p(\mathbf{z} \mid \mathbf{x}; \theta_t)$ we deduce:

$$\mathcal{L}(q_{t+1}(\mathbf{z}), \theta_t) = \ln p(\mathbf{x}; \theta_t),$$

$$\mathcal{L}(q_{t+1}(\mathbf{z}), \theta_{t+1}) \leq \ln p(\mathbf{x}; \theta_{t+1}).$$

Moreover, by definition of the M-step:

$$\mathcal{L}(q_{t+1}(\mathbf{z}), \theta_{t+1}) \geq \mathcal{L}(q_{t+1}(\mathbf{z}), \theta_t).$$

Putting all together we have:

$$\ln p(\mathbf{x}; \theta_{t+1}) \geq \mathcal{L}(q_{t+1}(\mathbf{z}), \theta_{t+1}) \geq \mathcal{L}(q_{t+1}(\mathbf{z}), \theta_t) = \ln p(\mathbf{x}; \theta_t).$$

Properties of the EM algorithm

- The log-marginal likelihood is **monotonically increasing**.
- The algorithm converges to a **stationary point** of the log-marginal likelihood.
- As the problem is generally not convex, the algorithm generally converges to a local optimum which strongly **depends on the initialization**.

EM algorithm summary

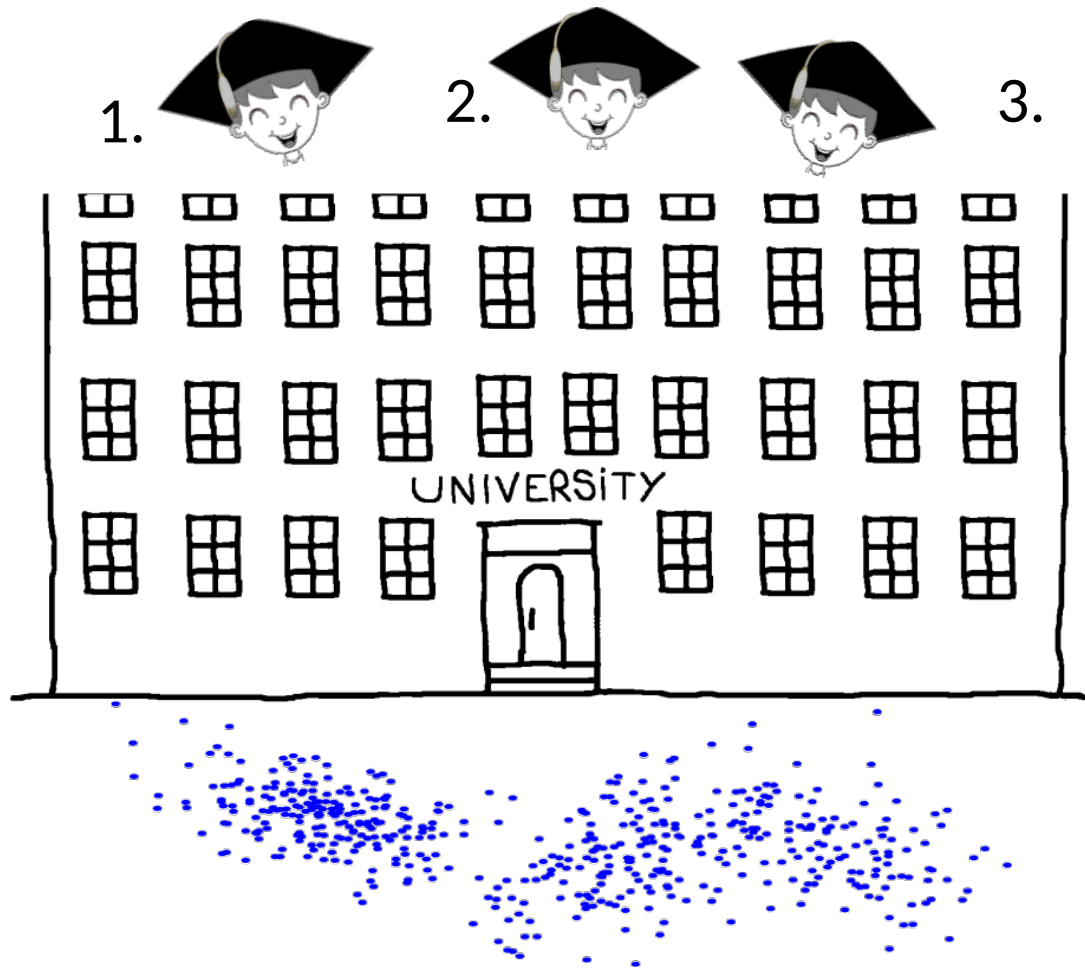
The EM algorithm can be reformulated in the space of the model parameters only.

Given an initialization θ_0 of the model parameters, iterate for $t = 0 : T - 1$:

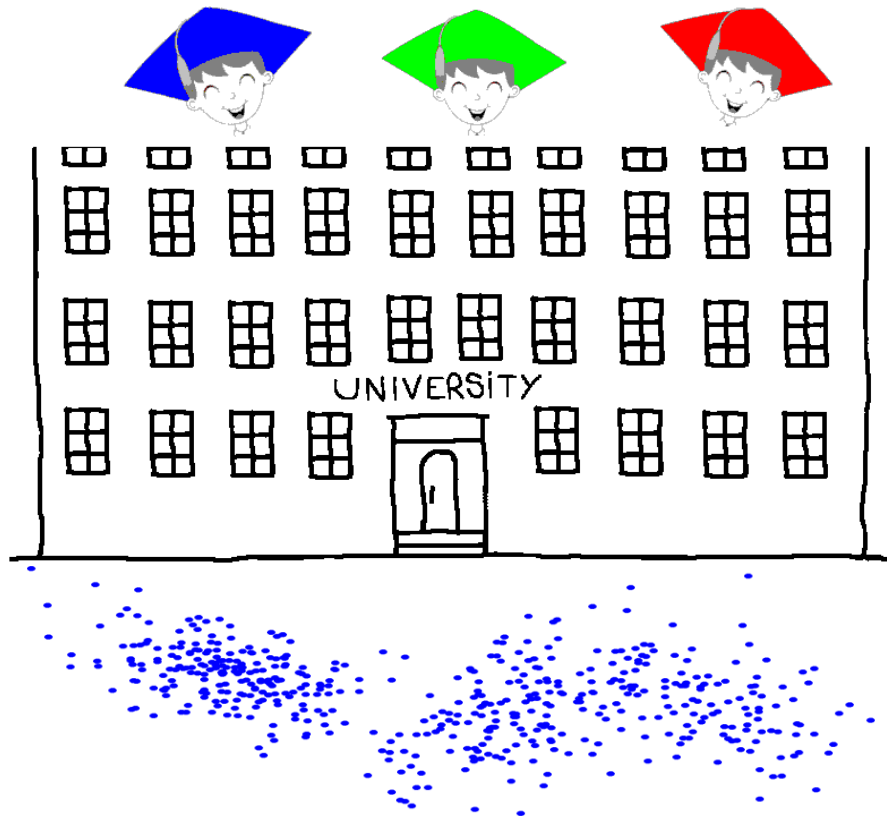
- **E-Step:** $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$;
- **M-Step:** $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$.

This is the recipe you should remember and use to derive an EM algorithm.

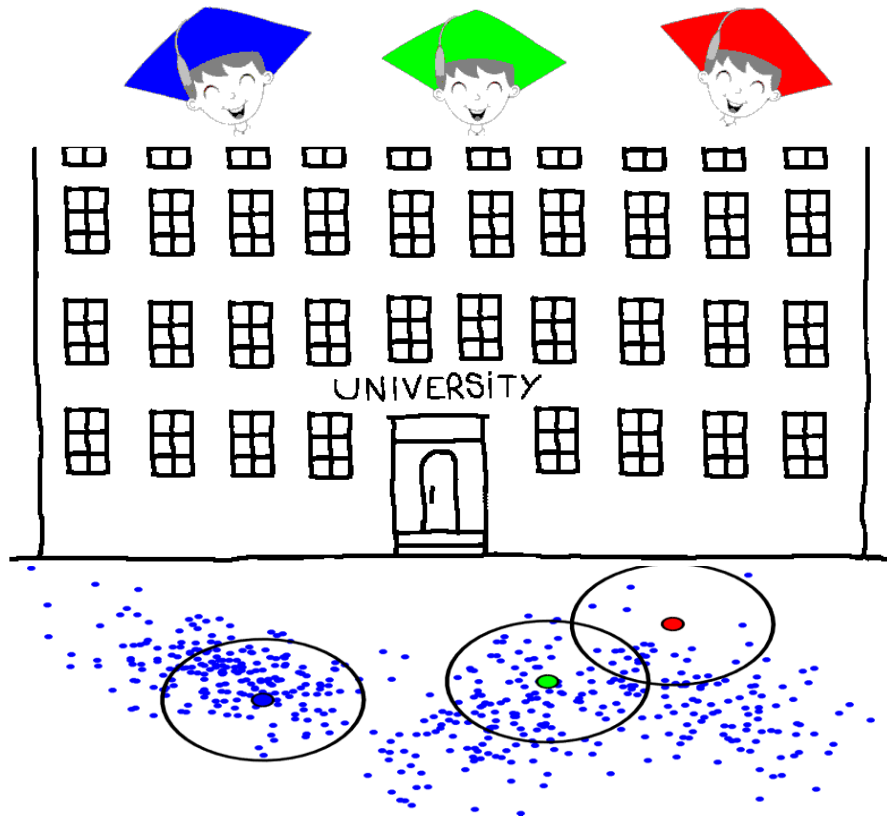
Back to the adventures of Thomas Bayes



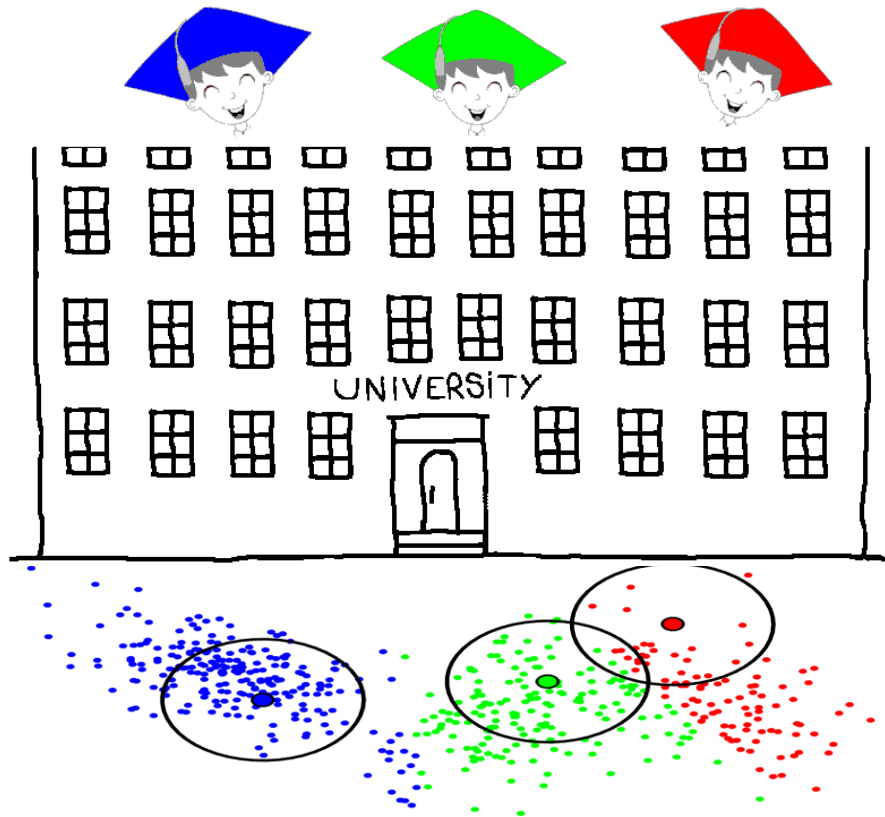
Let's derive an EM algorithm



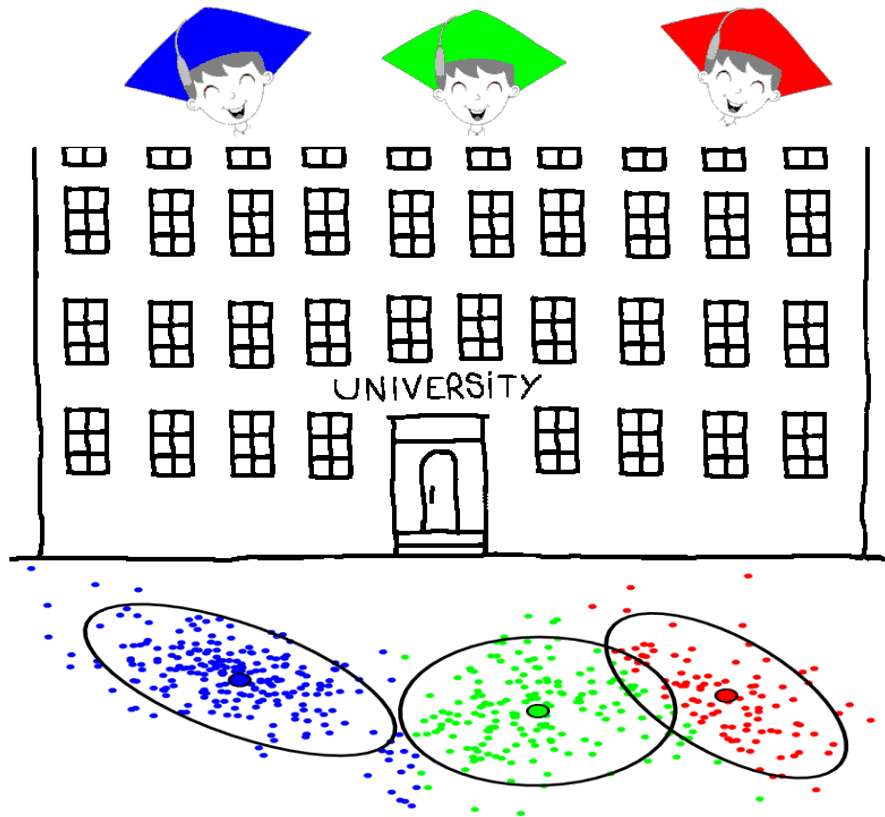
- Initialization: Random "guess" for θ_0
- E-Step: $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- M-Step: $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



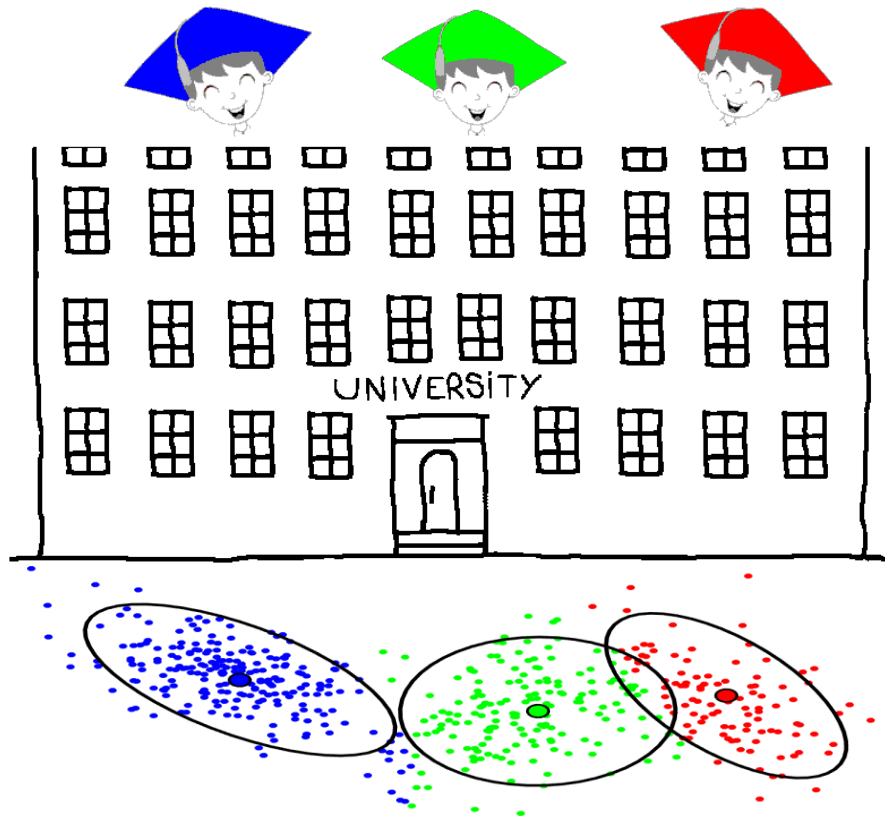
- **Initialization:** Random "guess" for θ_0
- E-Step: $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- M-Step: $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



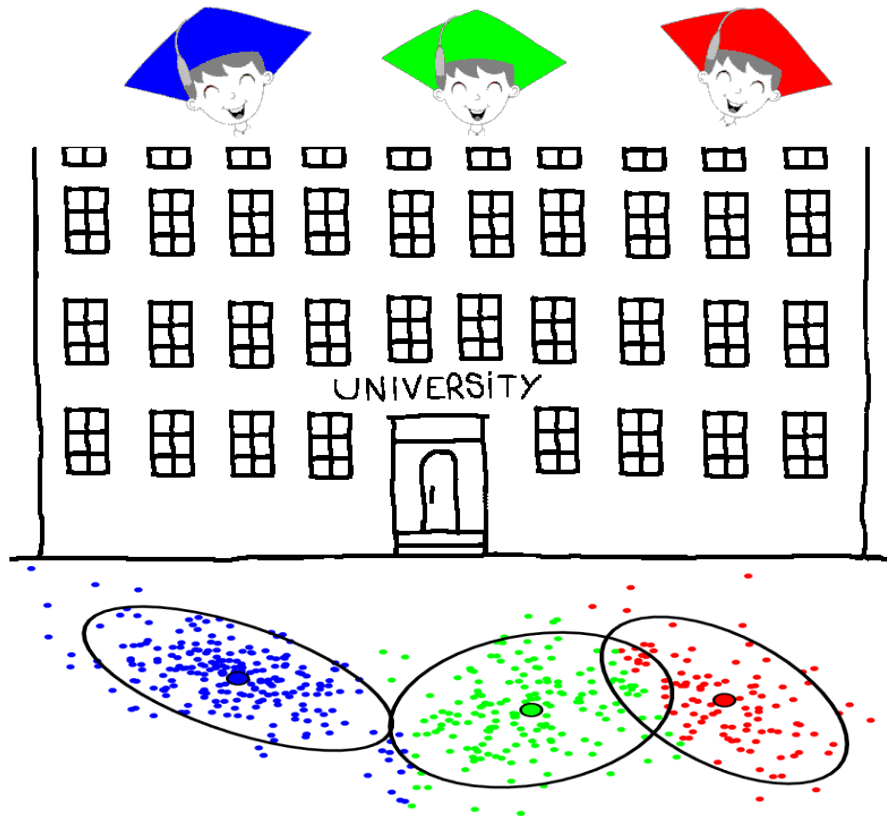
- Initialization: Random "guess" for θ_0
- **E-Step:** $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- M-Step: $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



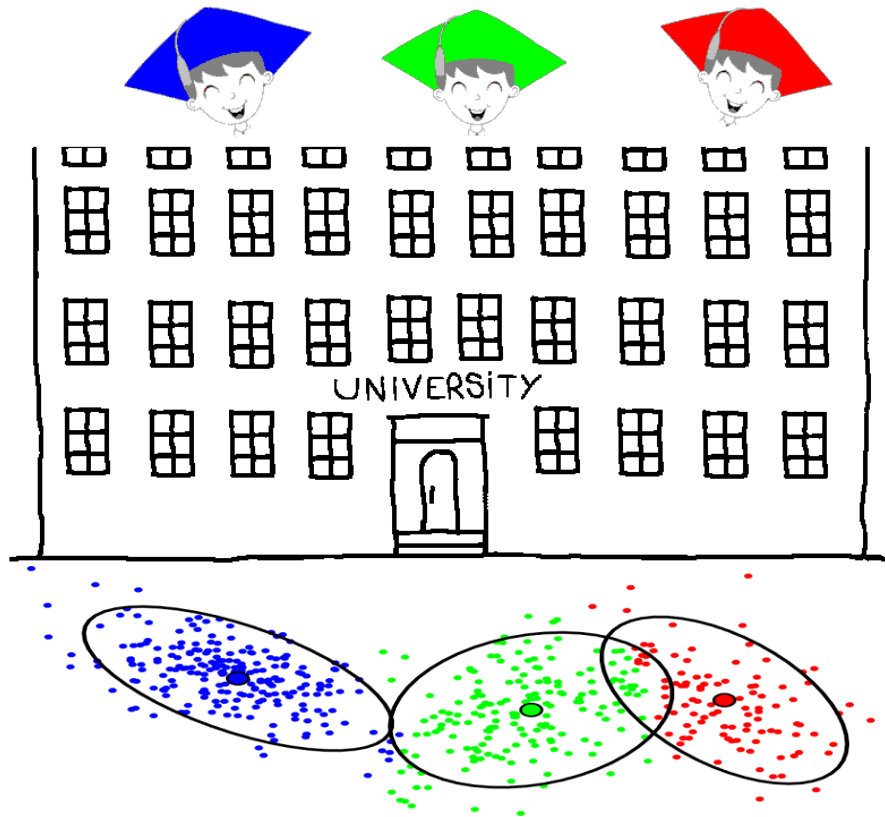
- Initialization: Random "guess" for θ_0
- E-Step: $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- **M-Step:** $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



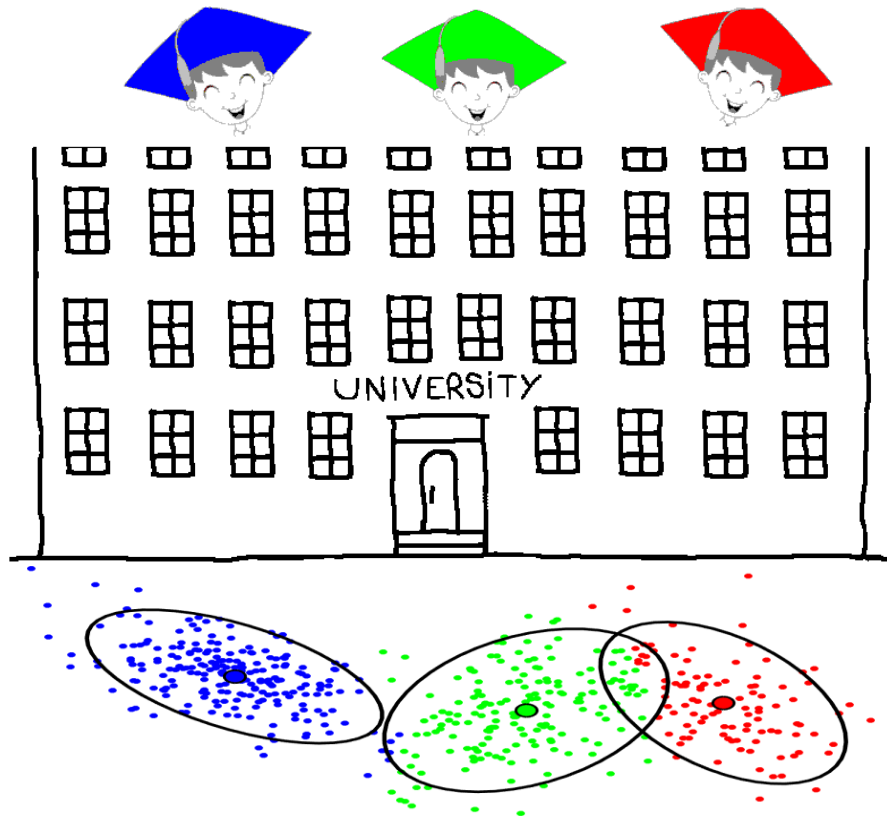
- Initialization: Random "guess" for θ_0
- **E-Step:** $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- M-Step: $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



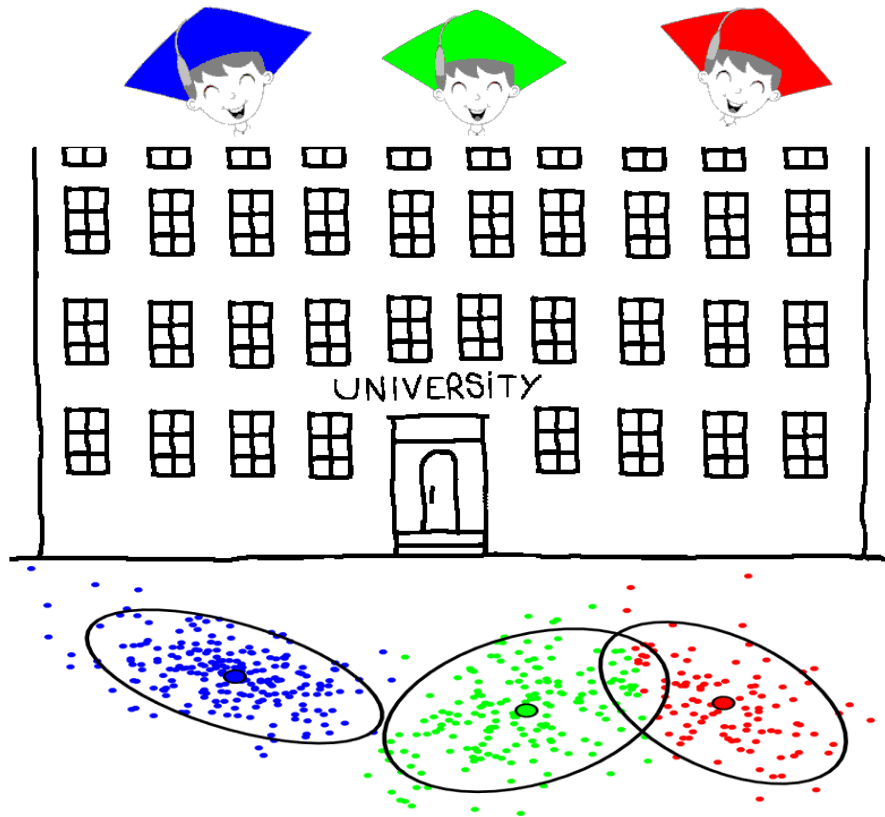
- Initialization: Random "guess" for θ_0
- E-Step: $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- **M-Step:** $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



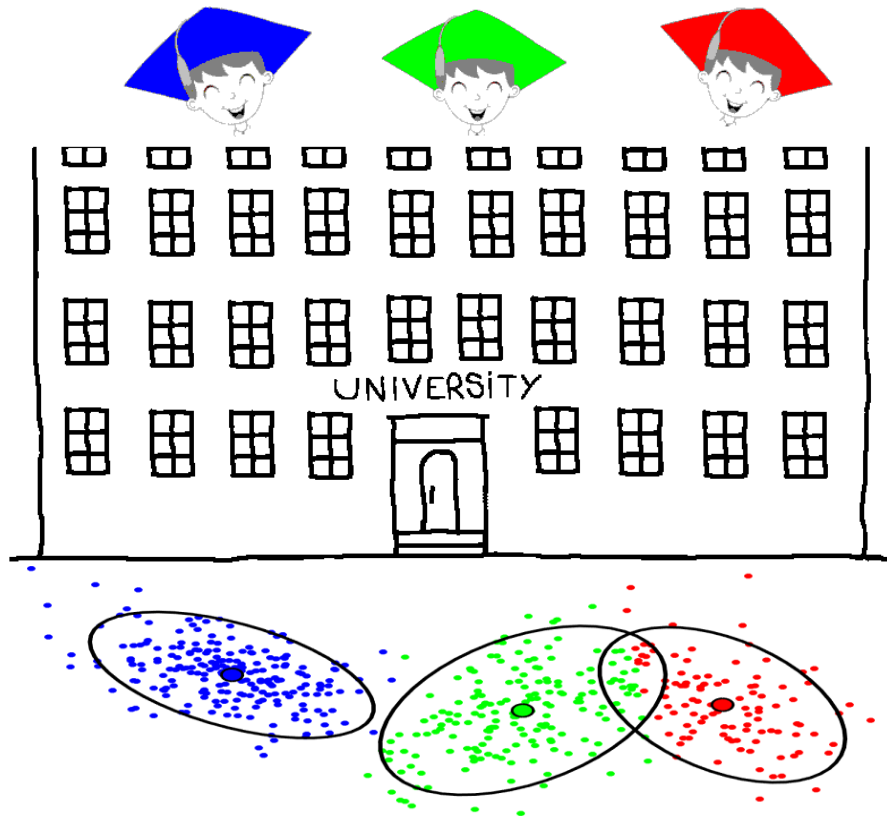
- Initialization: Random "guess" for θ_0
- **E-Step:** $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- M-Step: $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



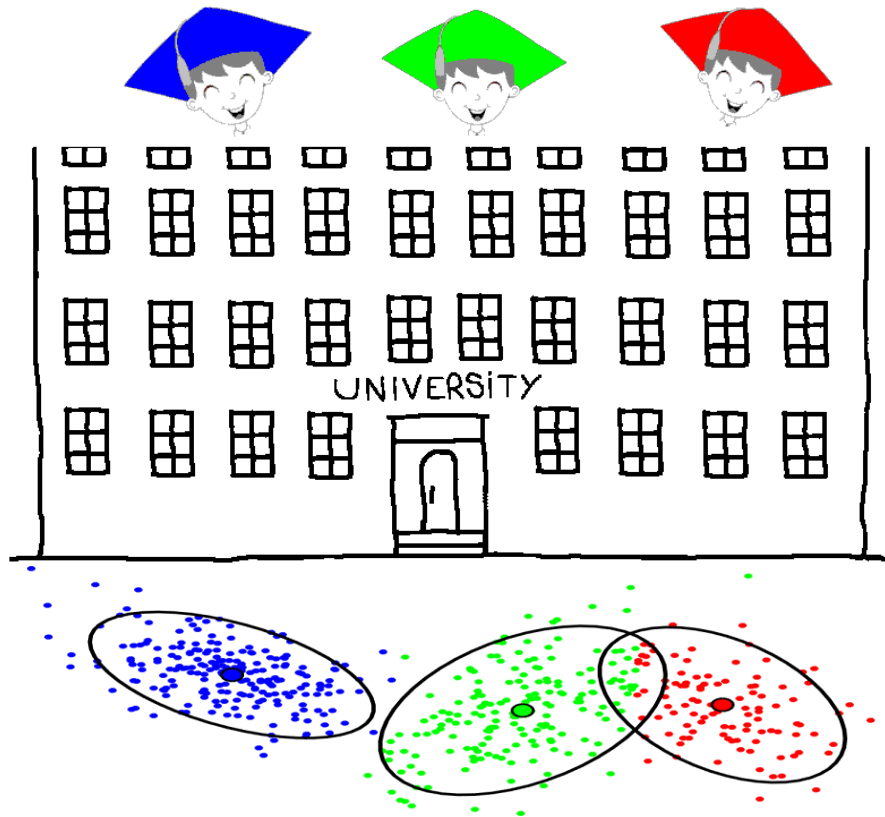
- Initialization: Random "guess" for θ_0
- E-Step: $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- **M-Step:** $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



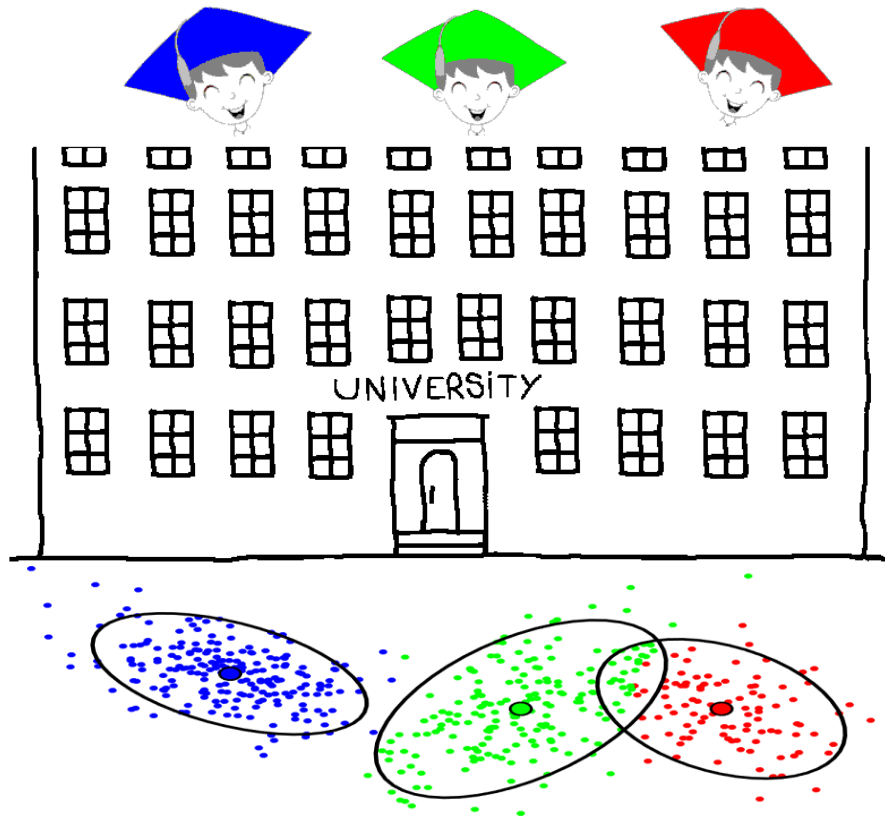
- Initialization: Random "guess" for θ_0
- **E-Step:** $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- M-Step: $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



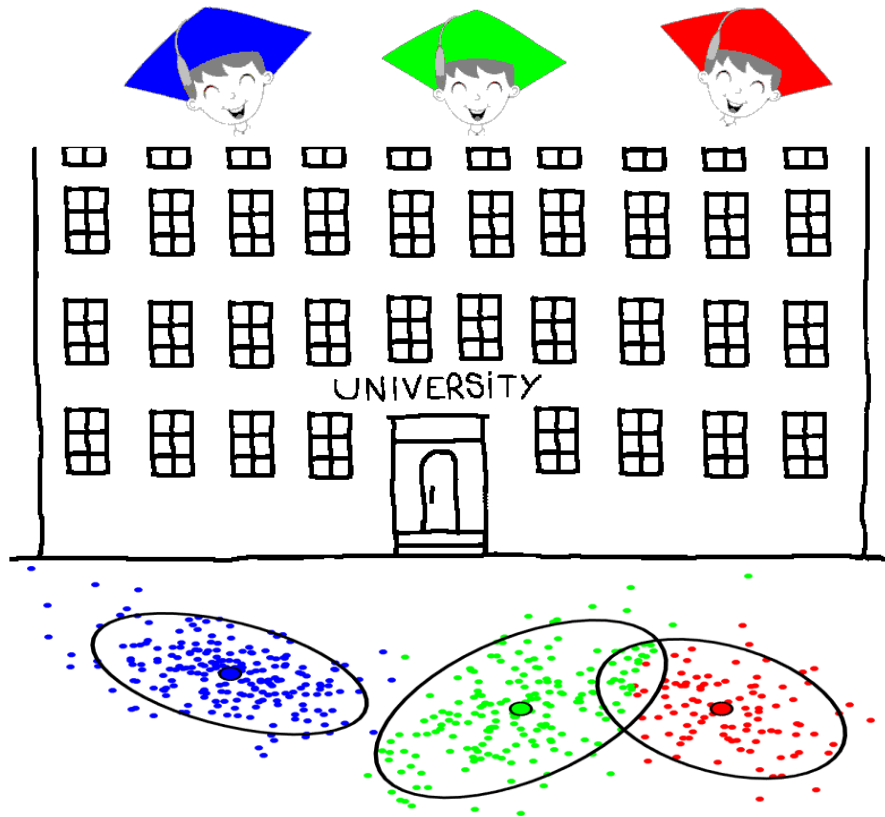
- Initialization: Random "guess" for θ_0
- E-Step: $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- **M-Step:** $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



- Initialization: Random "guess" for θ_0
- **E-Step:** $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- M-Step: $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- Convergence



- Initialization: Random "guess" for θ_0
- E-Step: $Q(\theta, \theta_t) = \mathbb{E}_{p(\mathbf{z}|\mathbf{x};\theta_t)}[\ln p(\mathbf{x}, \mathbf{z}; \theta)]$
- M-Step: $\theta_{t+1} = \arg \max_{\theta} Q(\theta, \theta_t)$
- **Convergence**



Minus 10 points for Mr. Green, minus 5 points for the others!

Lab session

- Theoretical work: Derivation of the EM algorithm for the GMM model.
- Practical work: Implementation of the EM algorithm.