MATHEMATICAL TOOLS FOR DATA SCIENCE

Leila GHARSALLI (leila.gharsalli@ipsa.fr)

IPSA, AERO 4

2023-2024

GOALS OF THE COURSE

- *Instructor : Leila GHARSALLI*
- *mail to : leila.gharsalli@ipsa.fr*
- Provide a general introduction to data mining and data science.
- Introduce some important tools for solving problems in data science.
- Grading : Participation in class (practical work) as well as a final exam will be graded.
- Main background needed: basic notions in probabilities and statistics, programming skill.

CONTENT OF THE COURSE

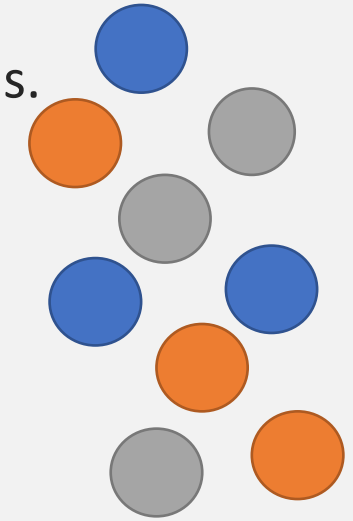
1. Introduction
2. Linear regression
3. Sparse regression
4. Classification
5. Principal Component Analysis
6. Clustering
7. Density estimation

INTRODUCTION

INTRODUCTION

Unsupervised learning: no knowledge of class output or value,

- Data Is unlabeled with a class and value are unknown,
- Goal: determine data patterns – groupings,
- Self-guided learning algorithm (k-means, genetic algorithms, clustering approaches…),
- Example: user behavior analysis.



?

INTRODUCTION

- **Unsupervised - Clustering**

INTRODUCTION

- Density estimation is the problem of estimating a probability distribution from data.

- As a first step we will introduce a probabilistic models for unsupervised learning.

INTRODUCTION

- We have a dataset without labels (we don't know anything about the population *a priori*). **Our goal is to learn something interesting about the structure of the data**:

- Clusters hidden in the dataset.

- Outliers: particularly unusual and/or interesting datapoints.

- Useful signal hidden in noise, e.g., human speech over a noisy phone.

INTRODUCTION

- We will assume that the dataset is sampled from a probability distribution P_{data}, which we will call the data distribution. We will denote this as

$$x \sim P_{data}$$

- The dataset $D = \{x^{(i)} \mid i = 1,2, \ldots, n\}$ consists of independent and identically distributed (IID) samples from P_{data}.

INTRODUCTION

- An unsupervised probabilistic model is a probability distribution

$$P(x): X \to [0,1].$$

This model can approximate the data distribution P_{data}.

- Probabilistic models also have parameters $\theta \in \Theta$, which we denote as

$$P_\theta(x): X \to [0,1]$$

INTRODUCTION

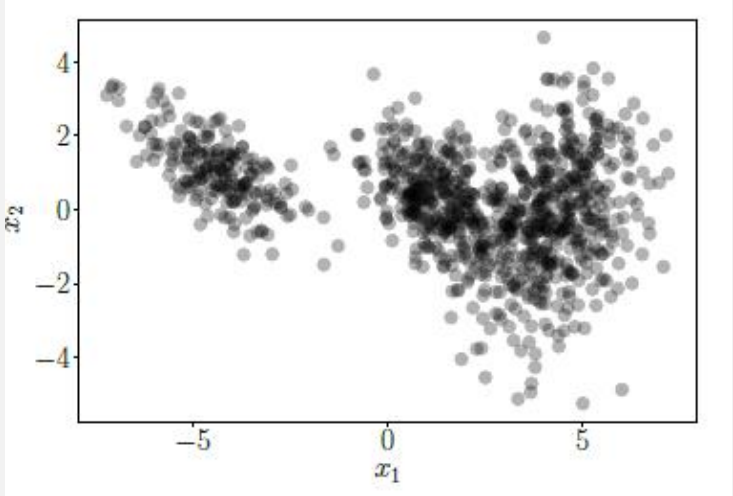
Why Use Probabilistic Models?

- There are many tasks that we can solve with a good model P_θ.
- Generation: sample new objects from P_θ, such as images.
- Representation learning: find interesting structure in P_{data}.
- Density estimation: approximate $P_\theta = P_{data}$ and use it to solve any downstream task (generation, clustering, outlier detection, etc.).
- We are going to be interested in the latter.

DENSITY ESTIMATION

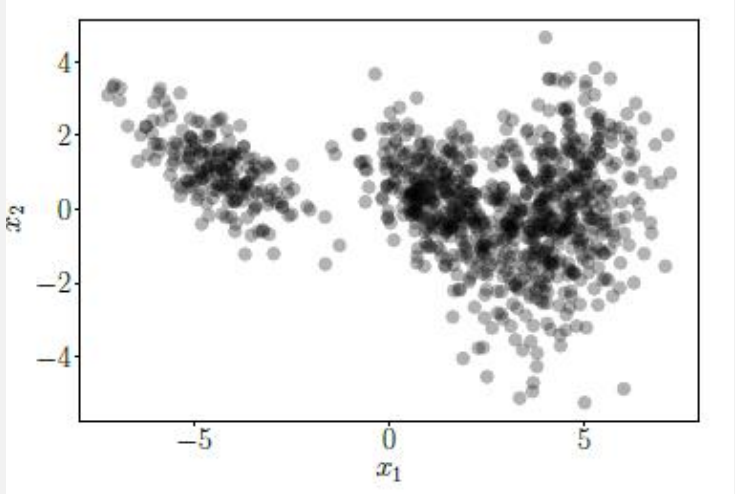
DENSITY ESTIMATION

- Density estimation: Given a dataset (unlabeled), find a probability density function from which the data could have plausibly been generated.
- Typically: Fix the class/model of densities and find optimal parameters given this class.
- Example. Class: Gaussian; Find mean and variance.
- **MLE/MAP estimation**.



DENSITY ESTIMATION

- Gaussians (or similarly all other distributions we encountered so far) have very limited modeling capabilities: Too simple.
- Mixture models are more flexible.



GAUSSIAN MIXTURE MODEL

FINITE MIXTURE MODEL

Given a data set $D = \{x_1, x_2, \ldots, x_N\}$ where x_i is a d-dimensional vector measurement. Assume that the points are generated in an IID fashion from an underlying density $p(x)$. We further assume that $p(x)$ is defined as a **finite mixture model** with K components:

$$\boldsymbol{p(x|\theta) = \sum_{k=1}^{K} \alpha_k \, p_k(x_k|z_k, \theta_k)}$$

Where:

- $p_k(x_k|z_k, \theta_k)$ are mixture components, $1 \leq k \leq K$. Each is a density or distribution defined over $p(x)$, with parameters θ_k.
- $z = (z_1, \ldots, z_K)$ is a vector of K binary indicator variables that are mutually exclusive and exhaustive.
- The $\alpha_k = p(z_k)$ are the mixture weights, representing the probability that a randomly selected x was generated by component k, where $\sum_{k=1}^{K} \alpha_k = 1$.

MEMBERSHIP WEIGHTS

- We can compute the "membership weight" of data point x_i in cluster k, given parameters θ as:

$$\boldsymbol{w_{ik} = p(z_{ik} = 1 | x_i, \theta) = \frac{p_k(x_k | z_k, \theta_k).\alpha_k}{\sum_{m=1}^{K} p_m(x_i | z_m, \theta_m).\alpha_m}, 1 \leq k \leq K, 1 \leq i \leq N}$$

- This follows from a direct application of Bayes rule.
- The membership weights above reflect our **uncertainty**, given x_i and θ, about which of the K components generated vector x_i . Note that we are assuming in our generative mixture model that each x_i was generated by a single component, so these probabilities reflect our uncertainty about which component x_i came from, not any "mixing" in the generative process.

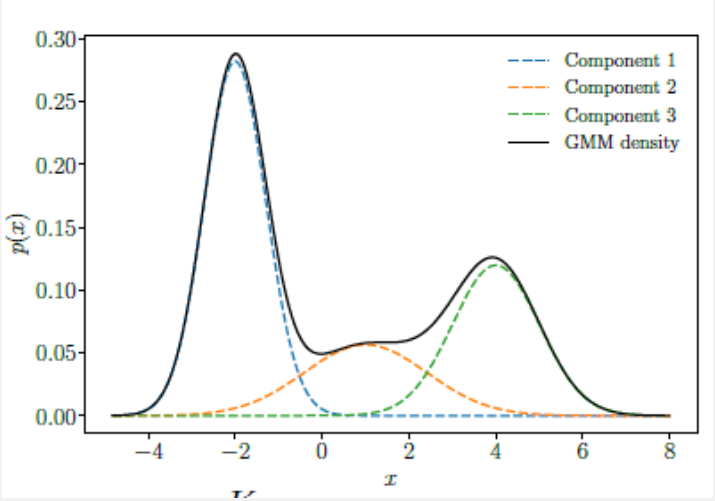
GAUSSIAN MIXTURE MODEL (GMM)

- A Gaussian mixture model is a density model where we combine a finite Gaussian mixture number of K Gaussian distributions $\mathcal{N}(x|\mu_k, \Sigma_k)$ so that:

$$\boldsymbol{p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}$$

$$\boldsymbol{0 \leq \pi_k \leq 1}$$

$$\boldsymbol{\sum_{k=1}^{K} \pi_k = 1}$$



- Individuel components are Gaussian distributions.
- Each component is weighted by π_k (mixture weights)

https://scikit-learn.org/stable/modules/mixture.html

PARAMETERS LEARNING

- Objective: Maximum likelihood estimate of model parameters $\boldsymbol{\theta}$ given a dataset $X;\ \theta = \{\pi_k, \mu_{k,} \Sigma_k, k = 1, \dots K\}$.

- Maximum Likelihood estimate:

$$\boldsymbol{\theta^* = argmax_\theta \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k) = argmax_\theta log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}$$

- Problem: difficult optimization problem (we can not move the log into the sum).
- Need for iterative scheme for learning the parameters.

GMM LIKELIHOOD

- Assume an i.i.d. data set $X = x_1, x_2, \ldots, x_N$ is given, and we want to determine the optimal parameters θ^* of the GMM via Maximum Likelihood.

- Likelihood

$$p(X|\theta) = \prod_{i=1}^{N} p(x_i|\theta);\ p(x_i|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)$$

- Log-likelihood

$$\log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)\right)$$

- **Learning objective: find the parameters θ^* that maximizes the log-likelihood.**

EXPECTATION-MAXIMIZATION ALGORITHM

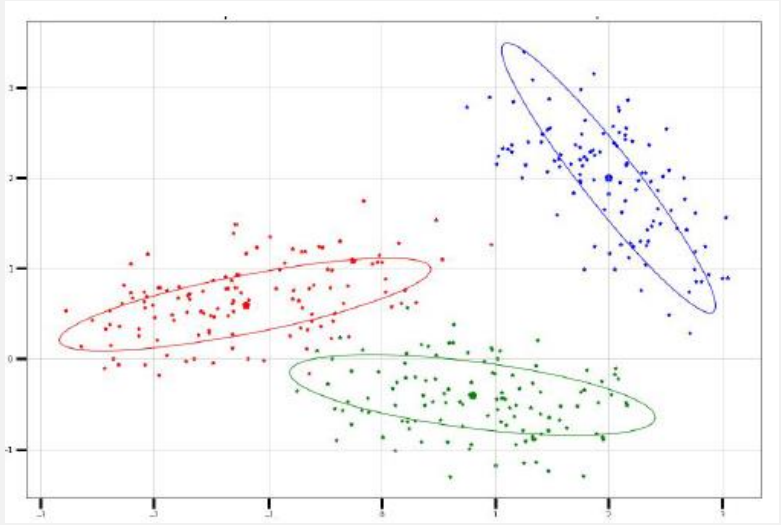
EM ALGORITHM

- Expectation-Maximization (EM) is a statistical algorithm for finding the right model parameters. We typically use EM when the data has missing values, or in other words, when the data is incomplete.

- These missing variables are called **latent variables**. We consider the target (or cluster number) to be unknown when we're working on an unsupervised learning problem.

EM ALGORITHM

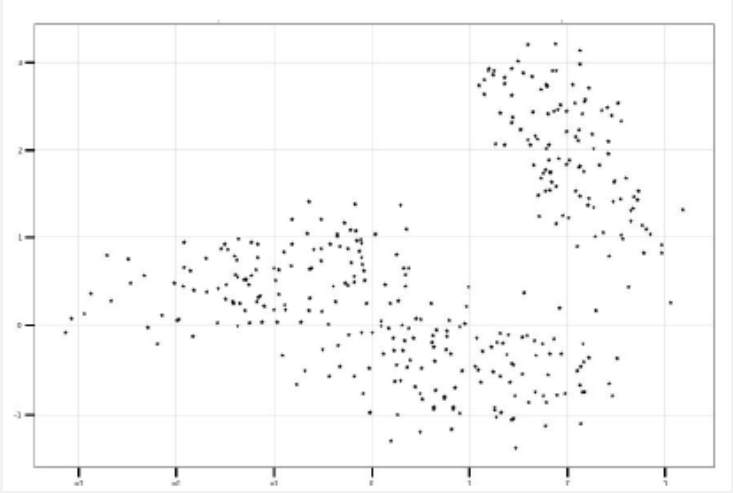
- Suppose we have $x_1, x_2, \ldots, x_n \in \mathbb{R}^2$ data points split in K different clusters coming from multivariate gaussian distributions.



K=3

EM ALGORITHM

- We don’t know to which cluster each point belongs to, we only know that there are K possibilities.
- We want an algorithm that allows us to find the **mean and covariance** of each cluster and guess the probability of membership for each data point.

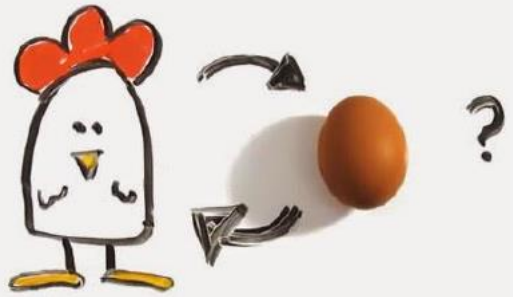


Clustering task

EM ALGORITHM

Chicken and egg problem

- If we knew the label of each point, we could estimate $\hat{\mu}$ and $\hat{\Sigma}$ for each cluster.
- If we knew the parameters μ and Σ of each cluster, we could compute the probability of membership for each class using the **Probability Density Function** of the gaussian distribution.



EM ALGORITHM

Solution : Start by a random initialization and iterate, hoping it converges toward the solution:

- We start with K randomly chosen gaussian distributions $(\pi_k, \mu_{k,} \Sigma_k)$,
- For each point, we compute the probability of membership for each class (Expectation part),
- Using the new membership probabilities that were just computed, we update the parameters of the gaussian distributions (Maximization of the likelihood part).
- **Theory on the EM algorithm ensures us that at each iteration, the expected value of the likelihood of the n-sample increases.**
- **For the convergence, check log-likelihood or the parameters.**

EM ALGORITHM

- **Initialize:** $\pi_k, \mu_{k,} \Sigma_k$
- **E-step:** compute probabilities for every data point x_i using current parameters $\pi_k, \mu_{k,} \Sigma_k$:

$$\boldsymbol{r_{ik}} = \frac{Probability\ of x_i\ to\ belong\ to\ a\ cluster\ C}{Sum\ of\ probabilities\ of x_i\ to\ belong\ to\ all\ clusters}$$

$$= \frac{\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)}{\sum \pi_j \mathcal{N}\left(x_i\middle|\mu_j, \Sigma_j\right)}$$

EM ALGORITHM

- **M-step:** Re-estimate parameters π_k, μ_k, Σ_k using the current probabilities r_{ik} (from E-step):

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} x_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} r_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

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

EXERCICE

In this exercise, we will use the unsupervised clustering by GMM on the iris data set.

1. Start by loading the iris dataset from the datasets package. Consider the only first two columns (sepal length and sepal width respectively).
2. Visualize your dataset.
3. Fit the data as a mixture of 3 Gaussians.
4. Assign a label to each observation (do the clustering).
5. Find the number of iterations needed for the log-likelihood function to converge and the converged log-likelihood value.
6. Print the converged log-likelihood value and the number of iterations needed for the model to converge.