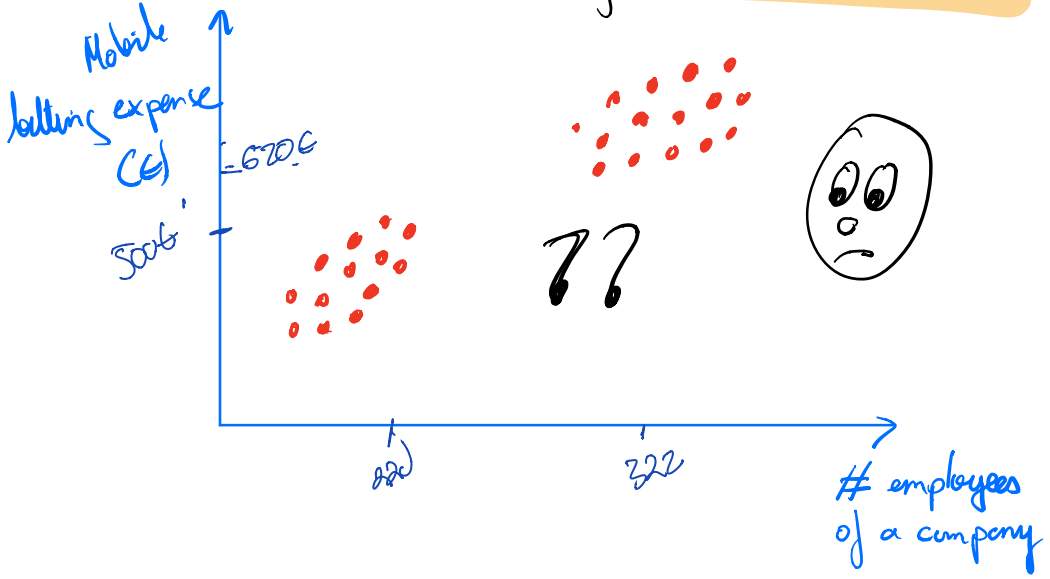
Introduction

From previous lessons, we know:

1. K-means don't work well with non-circular clusters
2. Gaussians models will allow to model our data with a Gaussian distribution for each cluster
   - means (μ) ⟷ center
   - covariance (Σ) ⟷ direction and shape

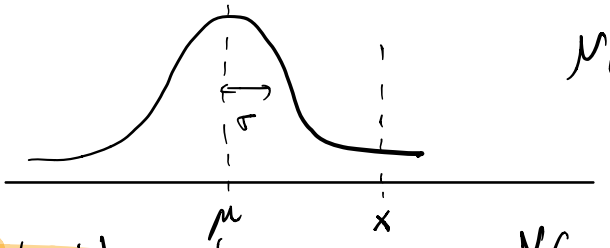
⇒ We have more flexible to capture clusters with different shapes and less sensitive to rescale of variables

⇒ Until now, we know to model 1 cluster. But, how to do it for several clusters??



Recall:

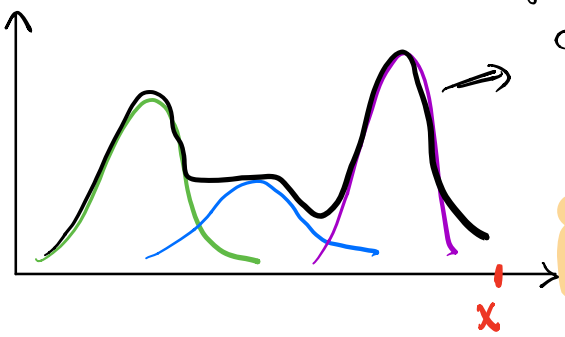
The Gaussian distribution in 1D is:



$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The probability of observing x given $\mathcal{N}(\mu,\sigma^2)$ is $p(x) = \mathcal{N}(x|\mu,\sigma)$ or likelihood

When the dataset "is organized" in clusters, we can consider that our data follows a complex function distribution and we can decompose in a superposition of Gaussians where:



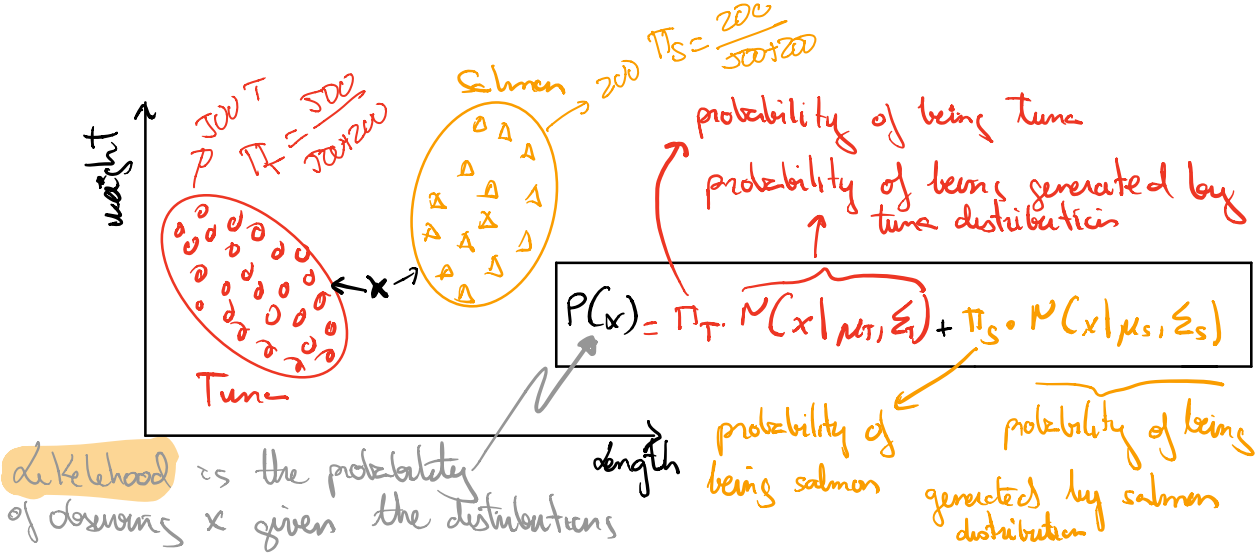
$$p(x) = \sum_{i=1}^{3} \pi_i \cdot \mathcal{N}(x|\mu_i,\sigma_i)$$

where π_k is prior probability that any datapoint is assigned to cluster k without observing the particular datapoint

where $\mathcal{N}(x,\mu_k,\Sigma_k)$ is the likelihood or the probability of observing x given a gaussian distribution k

An example:

Let's imagine a dataset with salmon and tuna. For a new datapoint x, the probability of observing x is:



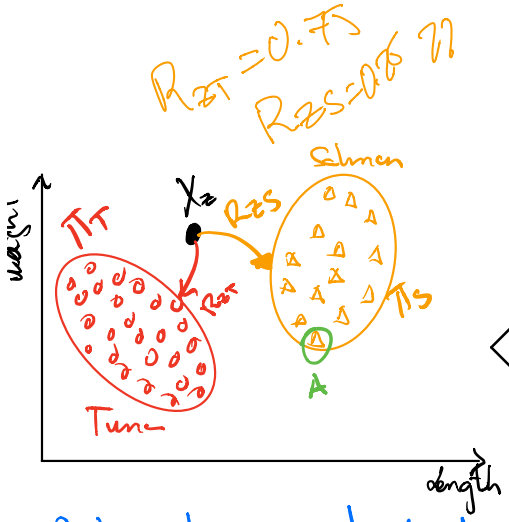
For the whole dataset $\vec{X}$ of N datapoints, where x_i is a datapoint with dimension D, the total likelihood function is:

$$p(\vec{X} \mid \pi, \mu, \Sigma) = \prod_{i=1}^{N} \left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \right)$$

As it's too computational cost, we calculate the log-likelihood:

$$\ln\left(p(\vec{X} \mid \pi, \mu, \Sigma)\right) = \sum_{i=1}^{N} \left[\left(\sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x_i \mid \mu_k, \Sigma_k) \right) \right]$$

The log-likelihood is the function to optimise in the training stage.



We calculate the derivative and we obtain:

$$\mu_K = \frac{1}{N_K} \sum_{i=1}^{N} R_{iK} \cdot x_i \quad , \quad N_K = \sum_{i=1}^{N} R_{iK}$$

$$\Sigma_K = \frac{1}{N_K} \cdot \sum_{i=1}^{N} R_{iK} (x_i - \mu_K)(x_i - \mu_K)^T$$

$$\pi_K = \frac{N_K}{N}$$

where R_{iK} is the responsability of datapoint i respect to cluster K; i.e R_{iK} is the posterior probability that a datapoint x_i is generated by component or cluster K:

Datapoints are not hard assigned to each cluster. Every datapoint, x_i, has a responsability or probability of being generated by each gaussian component.

$$\underbrace{p(K, x_i)}_{R_{iK}} = \frac{p(K) \cdot p(x_i|K)}{\sum_{l=1}^{K} p(l) \cdot p(x_i|l)} = \frac{\pi_K \cdot \mathcal{N}(x_i|\mu_K, \Sigma_K)}{\sum_{l=1}^{K} \pi_l \cdot \mathcal{N}(x_i|\mu_l, \Sigma_l)}$$

$\sum_{l=1}^{K} p(l) \cdot p(x_i|l) = p(x_i)$

Therefore, μ_K, Σ_K and π_K are weighted parameters according to datapoint's responsabilities

Therefore, our goal is to calculate the gaussian distributions, i.e. μ_K, Σ_K, π_K that give a maximum value of $p(\vec{X})$ for the whole dataset where:

$$\ln(p(\vec{X}|\pi, \mu, \Sigma)) = \sum_{i=1}^{N} \left[\left(\sum_{K=1}^{K} \pi_K \mathcal{N}(x_i|\mu_K, \Sigma_K) \right) \right]$$

is the log-likelihood of dataset $\vec{X}$

$$R_{2S} \quad p(S, A) = \frac{p(S) \cdot p(A|S)}{p(S) \cdot p(A|S) + p(T) \cdot p(A|T)}$$

But, what does it mean?

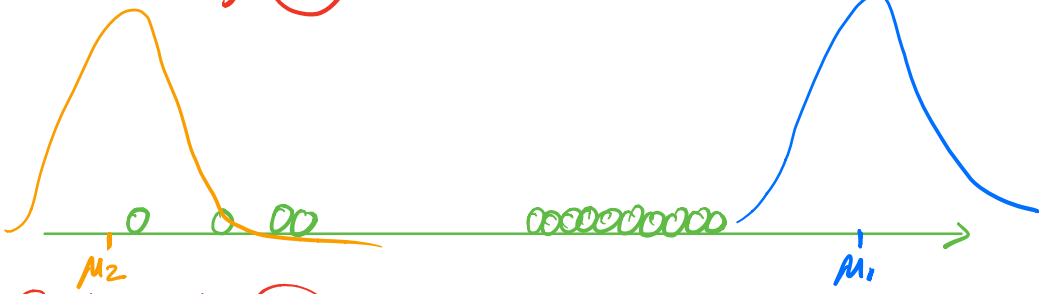
$p(T, A) = p(T) \cdot p(A|T)$

$p(T|p(A|T) + p(\bar{T}) \cdot p(A|\bar{T})$

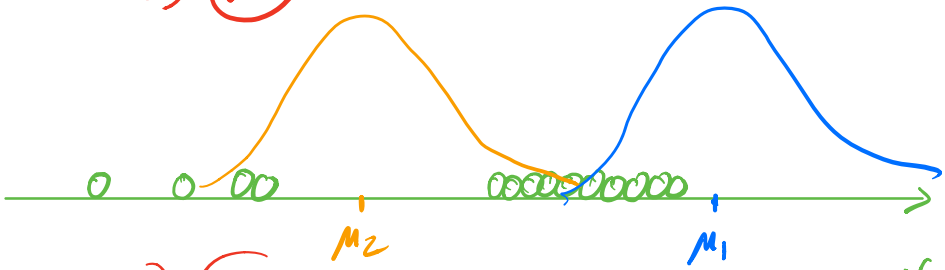
Imagine this 1-D datapoints distribution $P_{A,T}$

Which is the best Gaussian distributions??

Solution 1) (??)



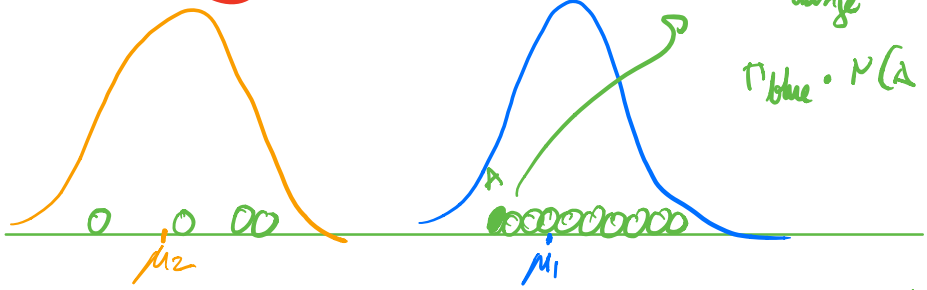
Solution 2) (??)



Solution 3) (??)

$p(A) = \pi_{orange} \cdot N(A | \mu_{orange}, \Sigma_{orange}) +$

$\pi_{blue} \cdot N(A | \mu_{blue}, \Sigma_{blue})$

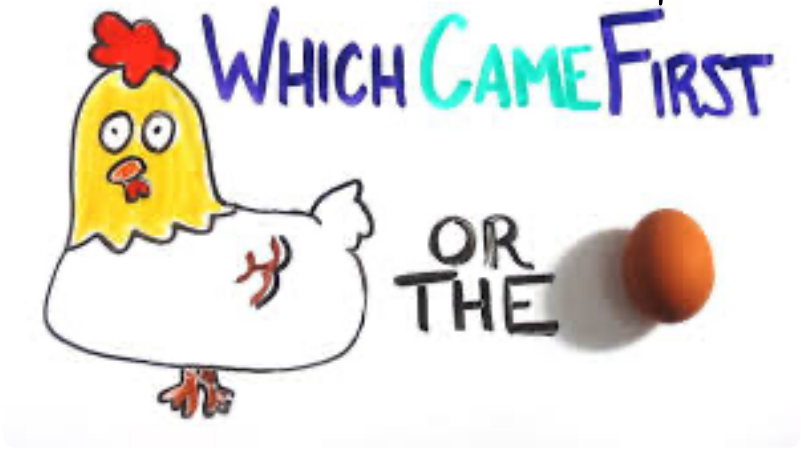


⇒ Gaussian distributions of solution 3) maximes the likelihood of the dataset because it maximes $p(x) = \sum_{k=1}^{2} \pi_k N(x | \mu_k, \Sigma_k)$ $x \in$ dataset

The Expectation - Maximization (EM) algorithm

The issue: When we have a new dataset X, we would to calculate μ_K, Σ_K and π_K for every cluster K. However, we need to know $N(X, \mu_K, \Sigma_K)$ of every cluster or component to calculate R_{iK} and to calculate $N(X, \mu_K, \Sigma_K)$ we need to know μ_K, Σ_K and π_K

We are in the EGG-CHICKEN problem



The solution:

Stage 0: We initialize randomly (or smarter) μ_K, Σ_K, π_K for every K

Stage 1: We calculate the membership or responsability of each datapoint x_i to every gaussian component (Estimation)

For each x_i, which its the probability to be generated by each component K $\Rightarrow$ R_{iK} for all K

Stage 2: We re-calculate or update each Gaussian based on the datapoints' responsabilities, R_{iK}: (Maximization)

For each K, calculate μ_K, Σ_K and π_K

Example 1D:

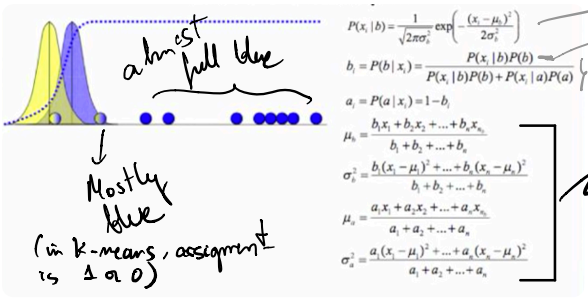
① Let's suppose this 1D dataset:

② Randomly, let's create two Gaussian functions ↔ random positions; i.e. μ_k / same σ

Likelihood ↓

1st step: Figure out how likely is that each point to come (E) from blue gaussian and yellow gaussian??

⇒ So, for each point we calculate $P(b|x_i)$; i.e. does x_i look like it come from b?

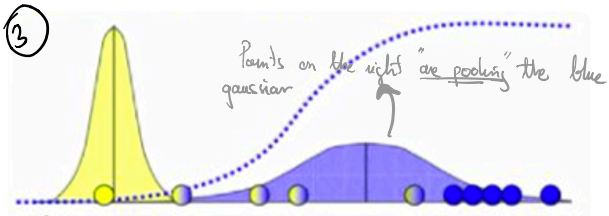


Probability x_i comes from b (what we know) → prior probability
→ Bayes posterior probability (gaussian distributions)

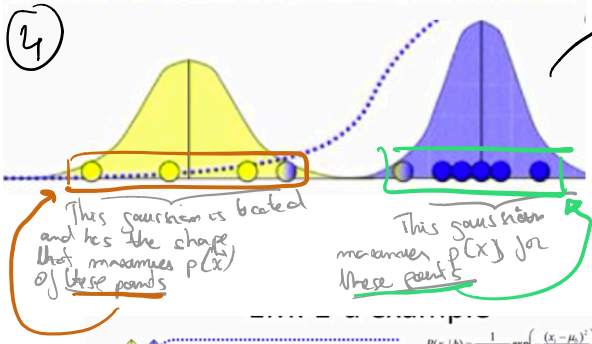
Thanks to previous probability, we can calculate $P(b|x_i)$

2nd step: Calculate the new μ_b, μ_y, Σ_b, Σ_y [M] with the datapoints' fraction assigned to each cluster

③



④



Following this process, we guarantee the optimization of the function of likelihood for the whole dataset $\vec{X}$; i.e. likelihood takes a maximum value

From the final picture, we see that we achieved a maximum value for $P(\vec{X}|\pi,\mu,\Sigma)$ due to gaussians are located at centers with Σ to maximize the value likelihood value for the whole dataset.

$$\ln\left(p(\vec{X}|\pi,\mu,\Sigma)\right) = \sum_{i=1}^{N}\left[\left(\sum_{k=1}^{K}\pi_k\, P(x_i|\mu_k,\Sigma_k)\right)\right]$$

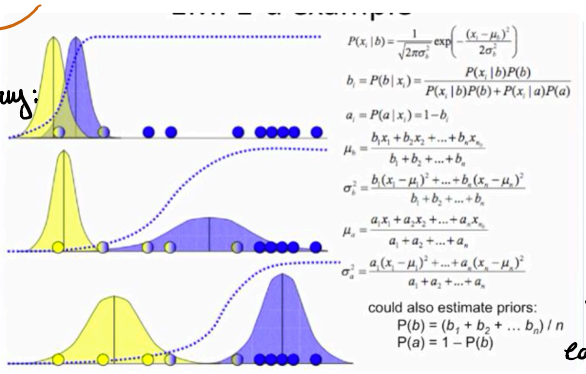
↳ probability of observing $\vec{X}$ given the distributions

Stage 0: Invent gaussian distributions for blue and yellow.

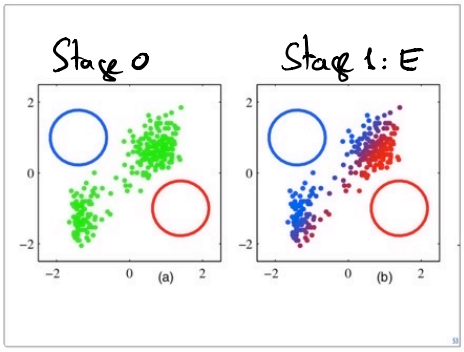
Stage 1: Estimate for each x_i their probability to come from blue a yellow distributions: $P(b|x_i)$ → by posterior pdb

Stage 2: Recalculate new distributions according to the probability of each x_i to come from each distribution

Summary:



Example 2-D:



Here we see EM in action. We start with two random Gausssians and color the points by how much responsibility each component takes. The blue points are mostly claimed by the blue component, the red points are mostly claimed by the red, and the purple points in the middle have the responsibility divided equally between the components.

source: Machine Leaerning and Pattern Recognition, Christopher Bishop.