Soft Clustering

Soft Clustering

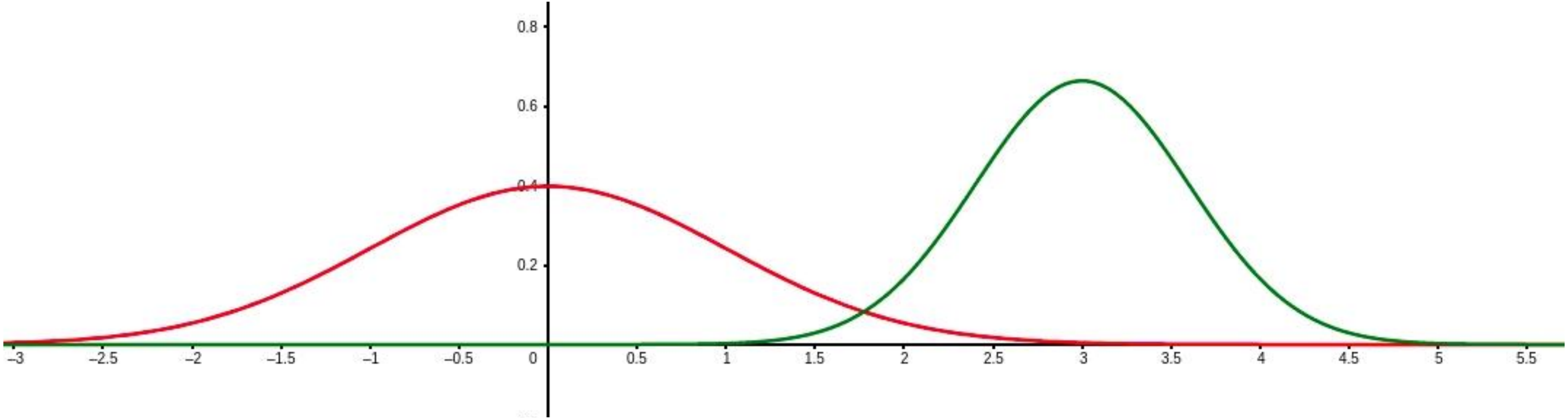
So far, clustering was done using **hard assignments** (1 point -> 1 cluster)

Sometimes this doesn't accurately represent the data: it seems reasonable to have overlapping clusters.

In this case, we can use **soft assignment** to assign points to every cluster with a certain probability.

Soft Clustering - Example

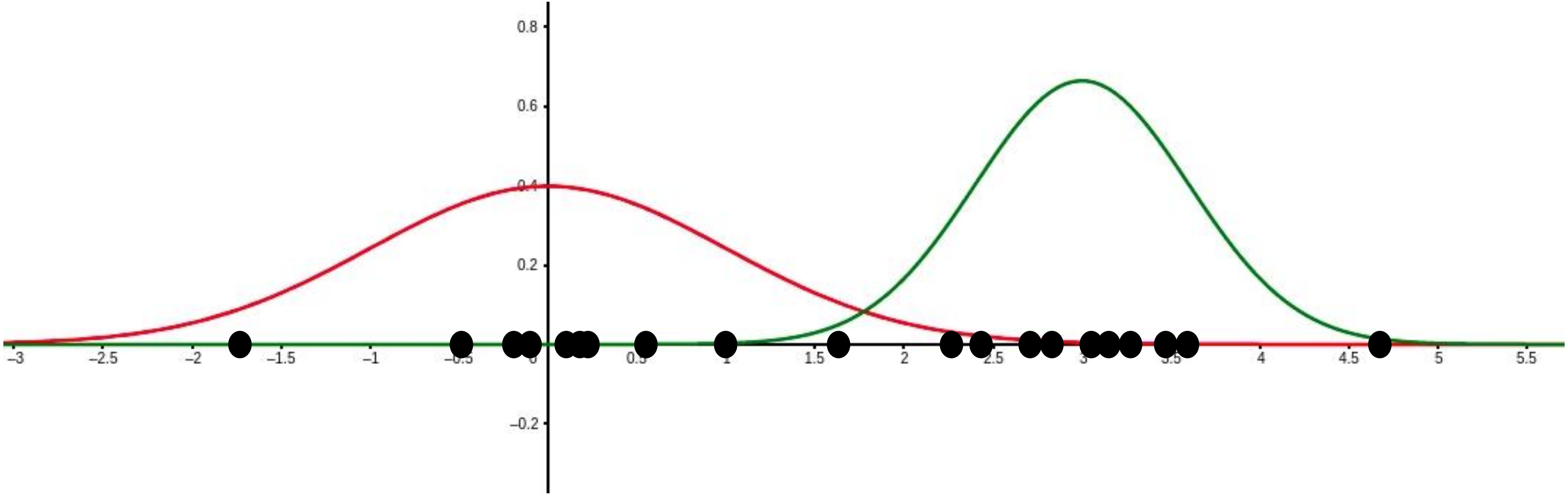
Generate data using $\mathbf{N(\mu_1, \sigma_1)}$ and $\mathbf{N(\mu_2, \sigma_2)}$



Or: we are given the weights of animals. Unknown to us these are weights from two different species. Can we determine the species (group / assignment) from the height?

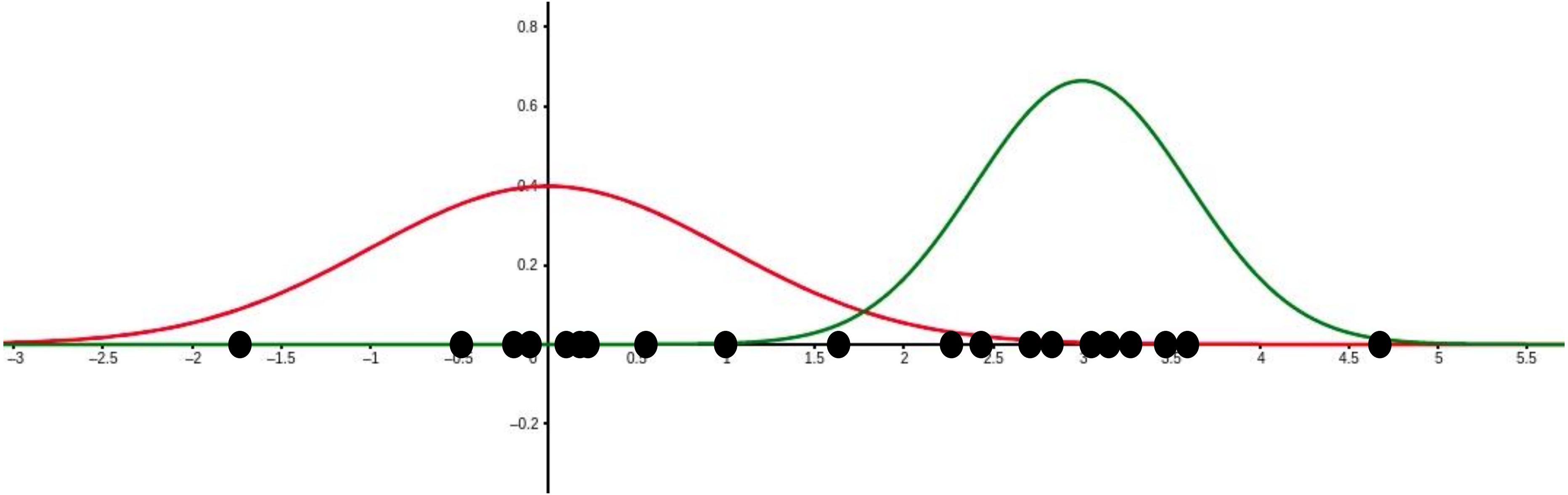
Soft Clustering - Example

Generate data using **N(μ_1, σ_1)** and **N(μ_2, σ_2)**



Soft Clustering - Example

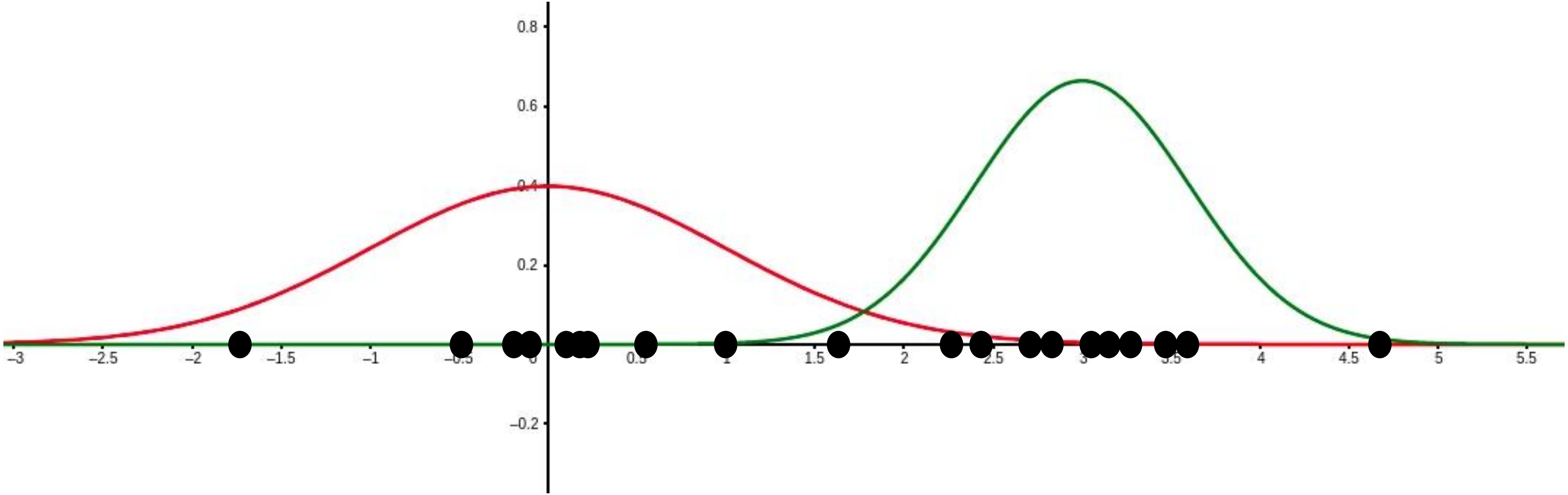
Generate data using **N(μ_1, σ_1)** and **N(μ_2, σ_2)**



Any of these points could technically have been generated from either curve.

Soft Clustering - Example

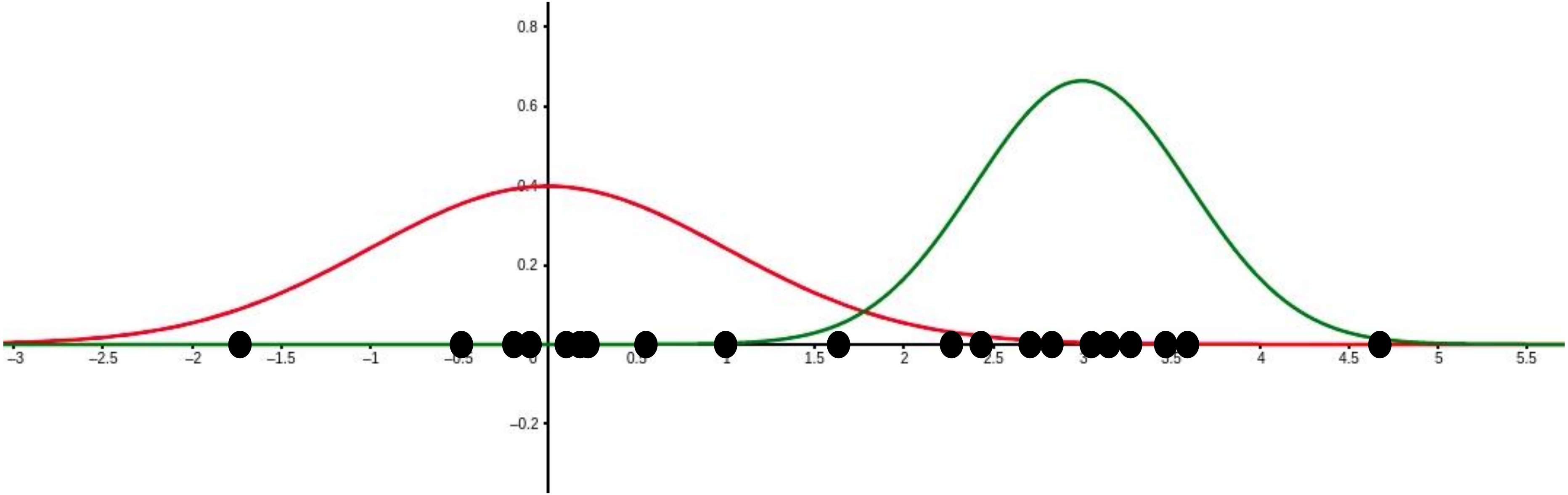
Generate data using $\mathbf{N(\mu_1, \sigma_1)}$ and $\mathbf{N(\mu_2, \sigma_2)}$



For each point we can compute the probability of it being generated from either curve

Soft Clustering - Example

Generate data using $\mathbf{N(\mu_1, \sigma_1)}$ and $\mathbf{N(\mu_2, \sigma_2)}$



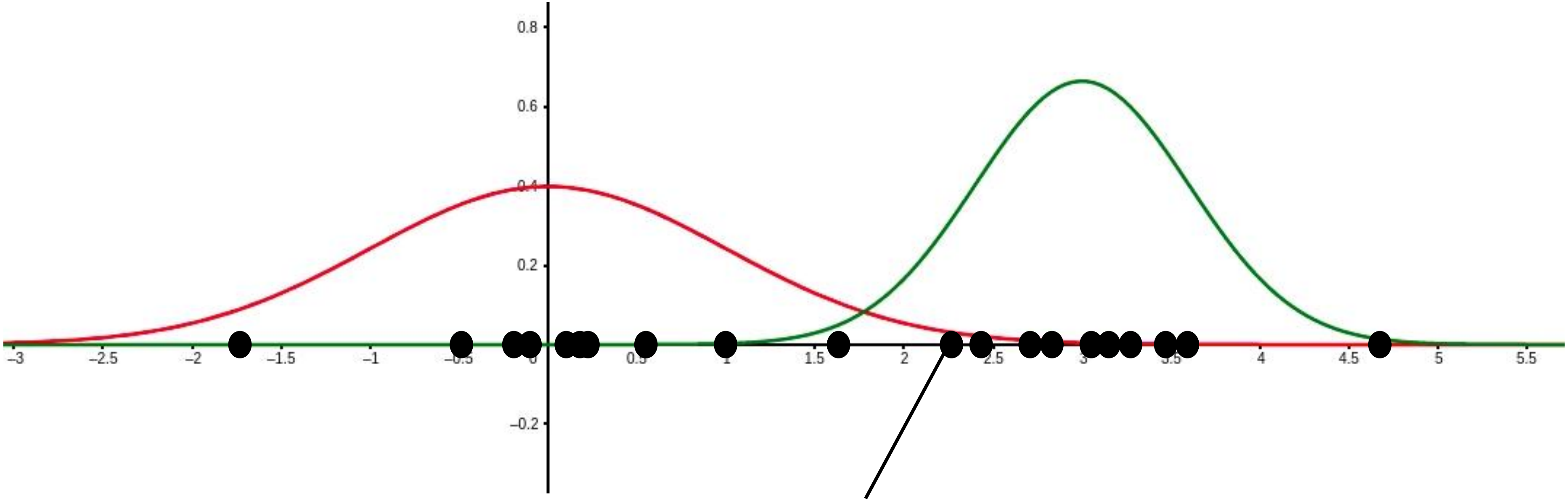
We can create soft assignments based on these probabilities.

Mixture Model

X comes from a mixture model with k mixture components if the probability distribution of X is:

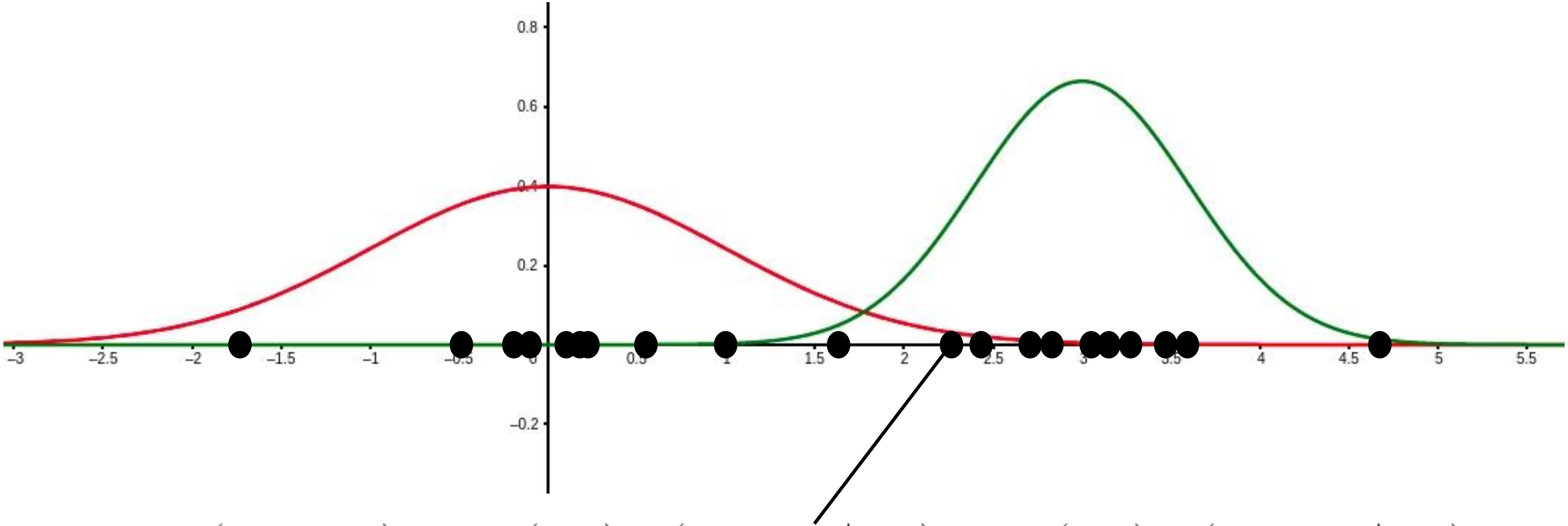
$$P(X = x) = \sum_{j=1}^{k} \underbrace{P(C_j)}_{\substack{\text{Mixture proportion} \\ \text{Represents the probability} \\ \text{of belonging to } C_j}} \underbrace{P(X = x|C_j)}_{\substack{\text{Probability of seeing x} \\ \text{when sampling from } C_j}}$$

Example



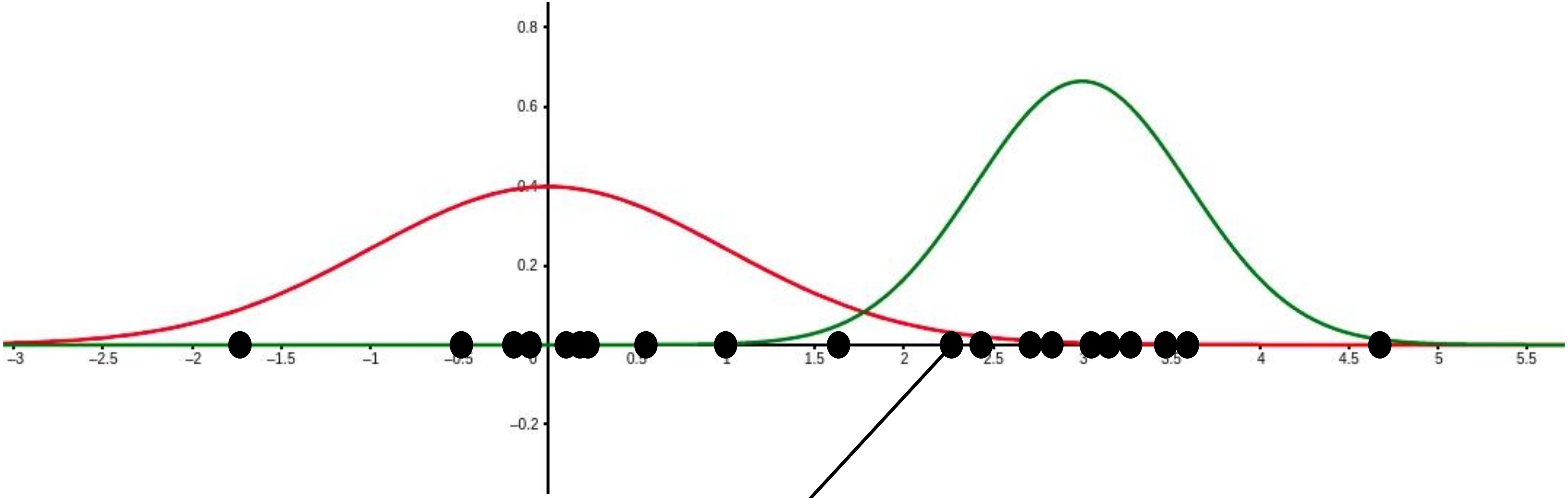
What is the probability distribution here?

Example



$$P(X = x) = P(C_1)P(X = x|C_1) + P(C_2)P(X = x|C_2)$$

Example



$$P(X=x) = P(C_1)\frac{1}{\sigma_1\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu_1}{\sigma_1}\right)^2} + P(C_2)\frac{1}{\sigma_2\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2}$$

Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a mixture model where

$$P(X = x | C_i) \sim N(\mu, \sigma)$$

GMM Clustering

Goal: Find the GMM that maximizes the probability of seeing the data we have.

The probability of seeing the data we saw is (assuming each data point was sampled independently) the product of the probabilities of observing each data point.
Finding the GMM means finding the parameters that uniquely characterize it.
What are these parameters?

$\mathbf{P(C_i)}$ & $\boldsymbol{\mu_i}$ & $\boldsymbol{\sigma_i}$ for all **k** components.

Lets call $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_k, P(C_1), \ldots, P(C_k)\}$

GMM Clustering

Goal:

$$\theta^* = \arg\max_{\theta} \prod_{i=1}^{n} \sum_{j=1}^{k} P(C_j) P(X_i \mid C_j)$$

Where $\boldsymbol{\Theta} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_k, P(C_1), \ldots, P(C_k)\}$

Joint probability distribution of our data

Assuming our data are independent

GMM Clustering

How do we find the critical points of this function?

Notice: taking the log-transform does not change the critical points

Define:

$$\begin{aligned} l(\theta) &= \log(L(\theta)) \\ &= \sum_{i=1}^{n} \log\left(\sum_{j=1}^{k} P(C_j)P(X_i \mid C_j)\right) \end{aligned}$$

GMM Clustering

For $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k]^T$ and $\boldsymbol{\Sigma} = [\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k]^T$

We can solve

$$\frac{d}{d\Sigma} l(\theta) = 0 \qquad \frac{d}{d\mu} l(\theta) = 0$$

GMM Clustering

To get

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} P(C_j|X_i)X_i}{\sum_{i=1}^{n} P(C_j|X_i)}$$

$$\hat{\Sigma}_j = \frac{\sum_{i=1}^{n} P(C_j|X_i)(X_i - \hat{\mu}_j)^T(X_i - \hat{\mu}_j)}{\sum_{i=1}^{n} P(C_j|X_i)}$$

$$\hat{P}(C_j) = \frac{1}{n}\sum_{i=1}^{n} P(C_j|X_i)$$

GMM Clustering

Do we have everything we need to solve this?

Still need $\mathbf{P(C_j \mid X_i)}$ (i.e. the probability that X_i was drawn from C_j)

GMM Clustering

$$P(C_j|X_i) = \frac{P(X_i|C_j)}{P(X_i)} P(C_j)$$
$$= \frac{P(X_i|C_j)P(C_j)}{\sum_{j=1}^{k} P(C_j)P(X_i|C_j)}$$

Looks like a loop! Seems we need $P(C_j)$ to get $P(C_j \mid X_i)$ and $P(C_j \mid X_i)$ to get $P(C_j)$

Expectation Maximization Algorithm

1. Start with random **θ**
2. Compute $\mathbf{P(C_j \mid X_l)}$ for all $\mathbf{X_i}$ by using **θ**
3. Compute / Update **θ** from $\mathbf{P(C_j \mid X_l)}$
4. Repeat 2 & 3 until convergence

Demo