Gaussian Mixture Model

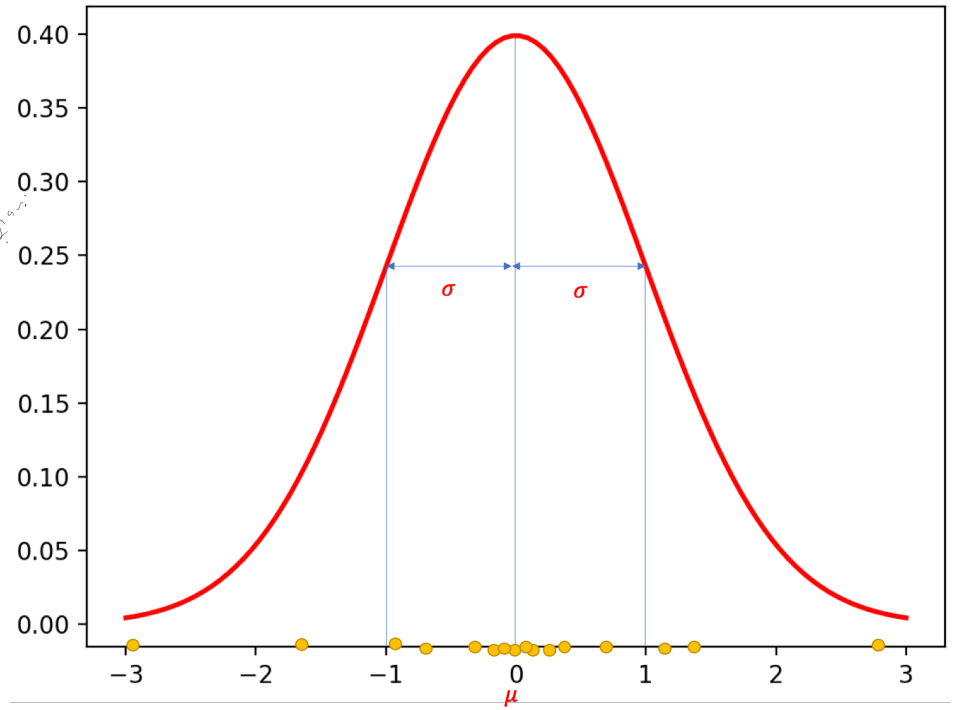
Sujay Sanghavi

Gaussian Mixture Model

A probability model for clustered data.

Recall: Single Gaussian 1-dimension:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



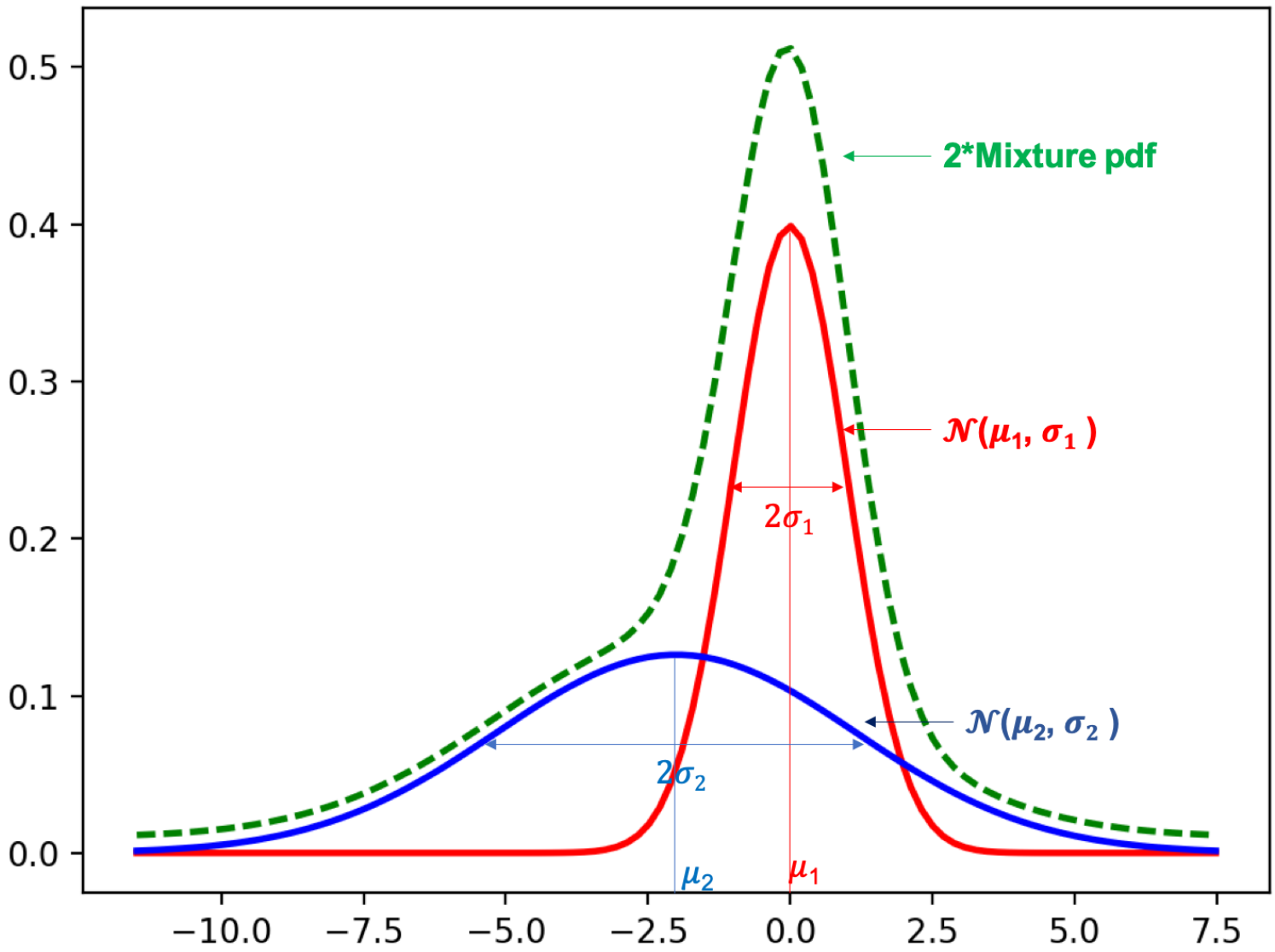
Gaussian Mixture Model

n sample points generated i.i.d as follows:

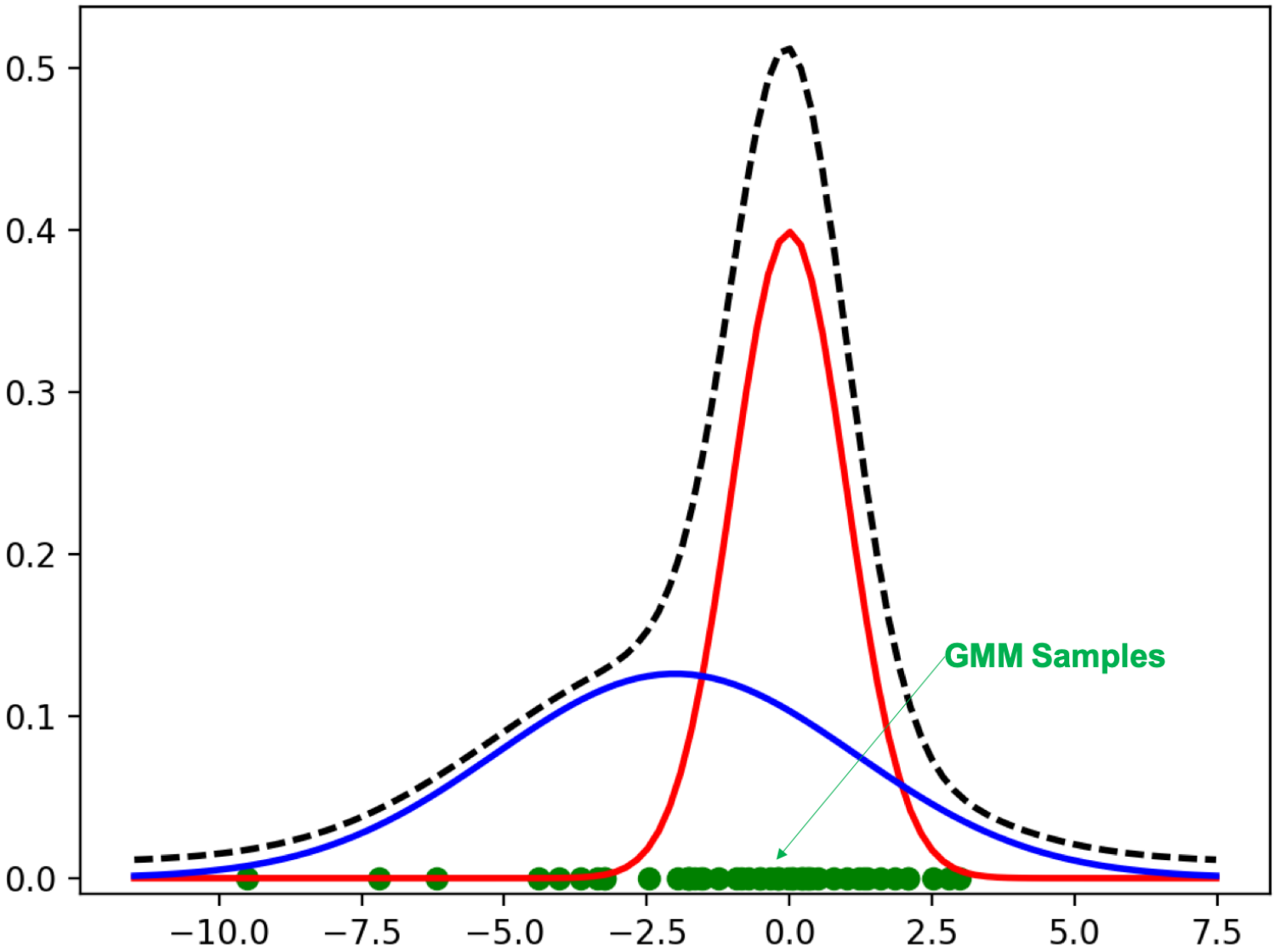
- Pick Gaussian i with probability p_i
- Generate a point from $\mathcal{N}(\mu_i, \sigma_i)$

TASK: Given these (unlabeled) points, find the k Gaussians

Gaussian Mixture Model

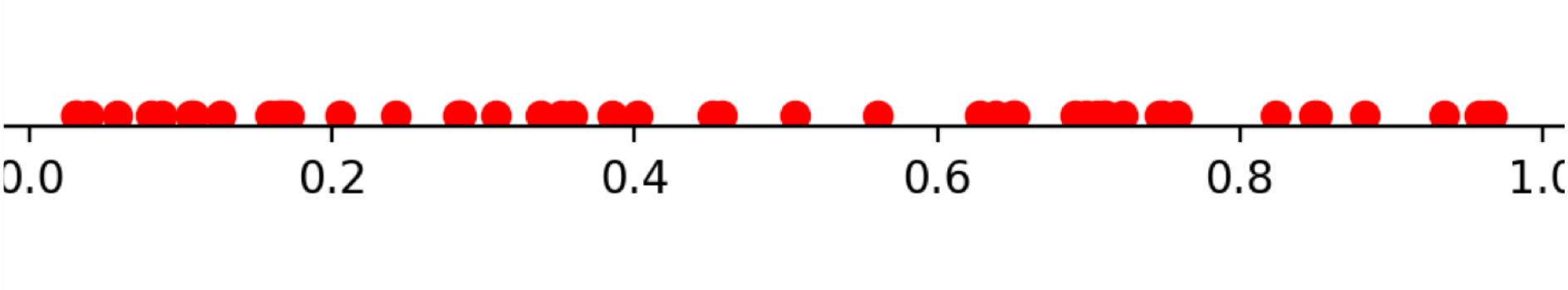


Gaussian Mixture Model



Gaussian Mixture Model

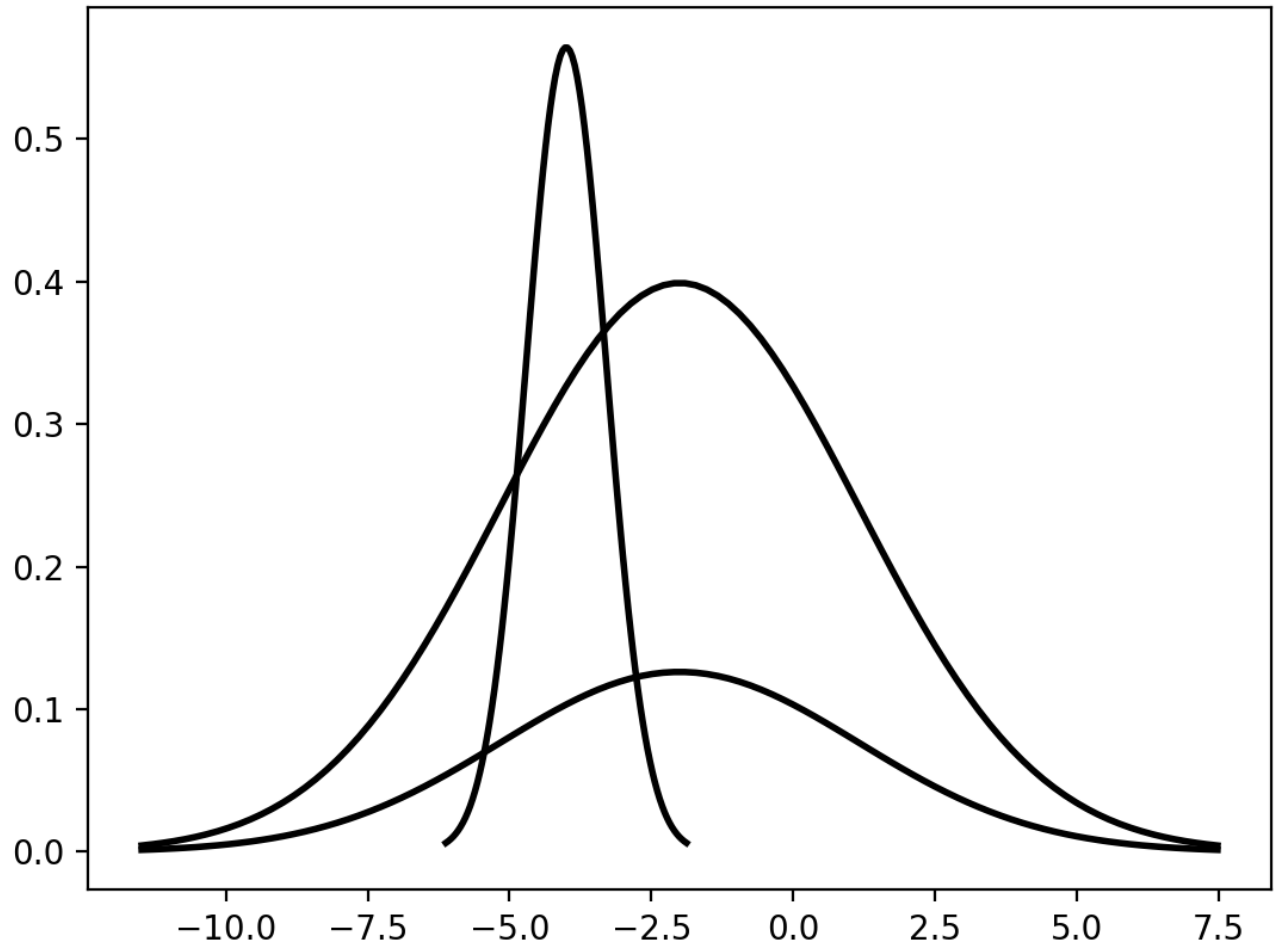
TASK: Given only the samples, find the k Gaussians



Red points: GMM Samples

Gaussian Mixture Model

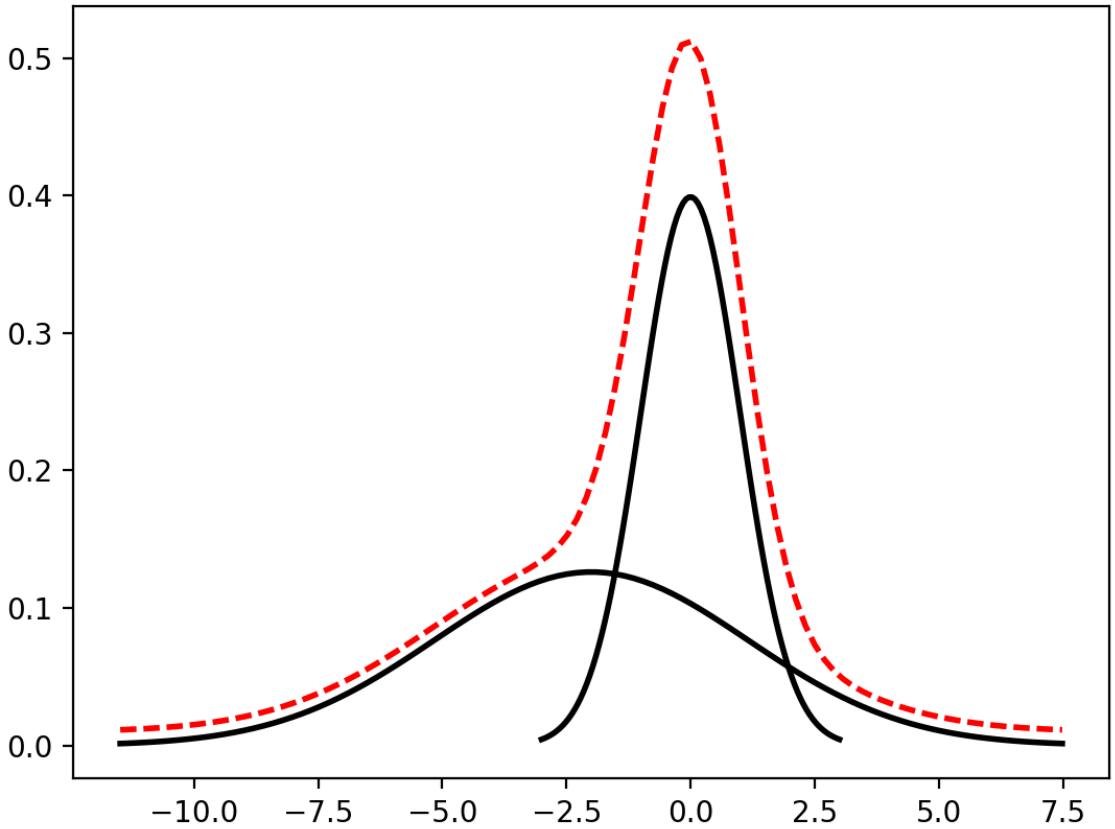
k Gaussians



Gaussian Mixture Model

Each $\mathcal{N}(\mu_i, \sigma_i)$ also has a probability p_i

$$\sum_i p_i = 1. \text{ Here, } p_1 = 3/4, p_2 = 1/4$$



Gaussian Mixture Model

Recall Task:

Given $x_1, \cdots x_n$ from a mixture of k Gaussians

Find parameters of GMM

$$\begin{aligned} \text{means} \quad & \mu_1, \cdots, \mu_k \\ \text{variances} \quad & \sigma_1^2, \cdots, \sigma_k^2 \\ \text{prior probabilitiies} \quad & p_1, \cdots, p_k \end{aligned}$$

Likelihood for a Single Gaussian

Consider a datapoint x from a Gaussian.

$$p(x) = L(x;\mu) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right)$$

For n datapoints

$$L(x_1,\ldots,x_n;\mu) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_j-\mu)^2}{2}\right)$$

Now maximum likelihood of μ is

$$\begin{aligned}
\hat{\mu_{ML}} &= \arg\max_{\mu} L(x_1,\ldots,x_n;\mu) \\
&= \arg\max_{\mu} \log L(x_1,\ldots,x_n;\mu) \\
&= \arg\max_{\mu} \sum_j \left[-\frac{(x_j-\mu)^2}{2}\right] \\
&= \frac{\sum_j x_j}{n}
\end{aligned}$$

Likelihood

Now for our task of n datapoints from k Gaussians:

$$L(\{x_j\}; \{\mu_i, \sigma_i^2, p_i\}) = \prod_{j=1}^{n} \left\{ \sum_i p_i \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(x_j - \mu_i)^2}{2\sigma_i^2} \right) \right\}$$

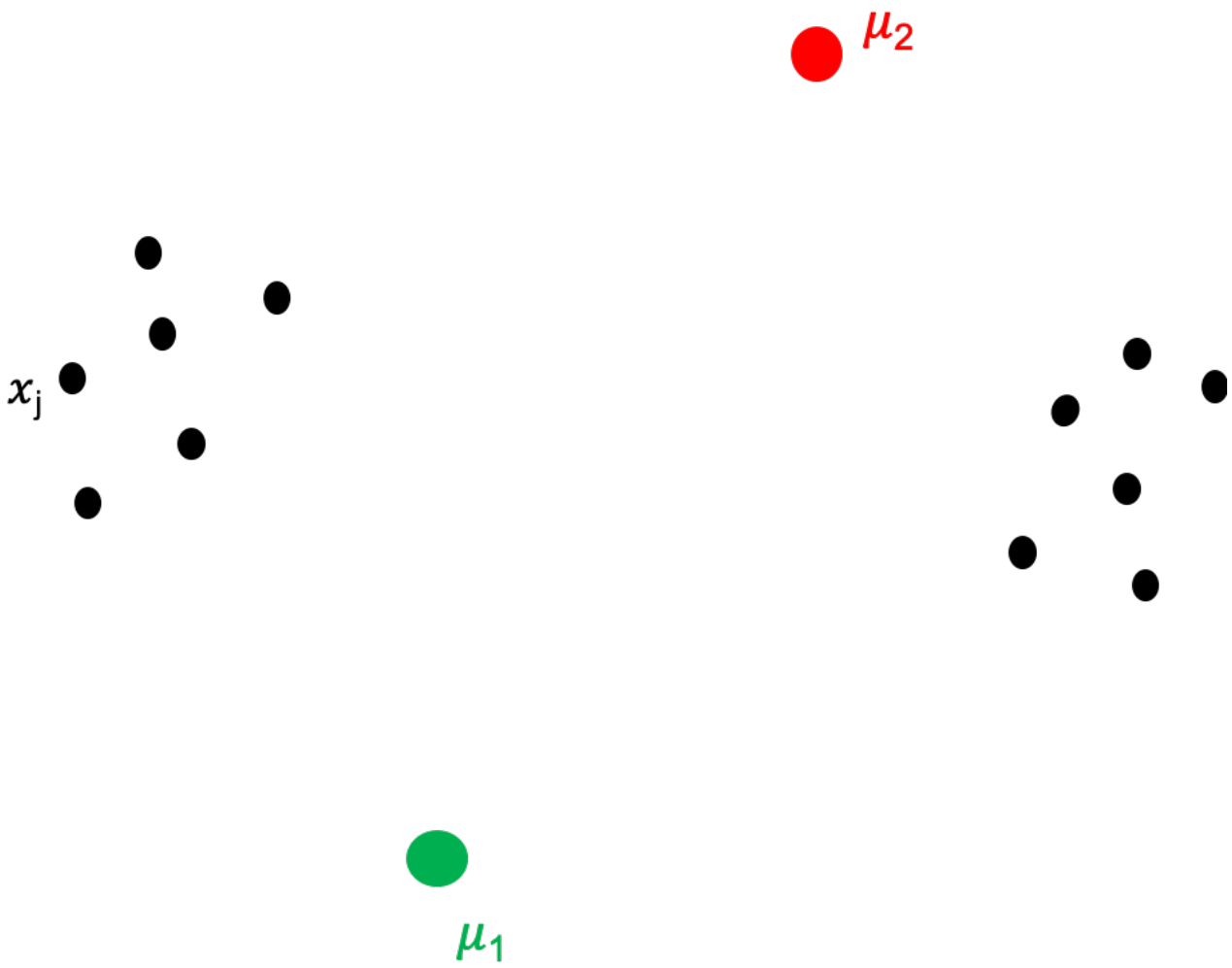
Maximum likelihood will be

$$\arg \max_{\{\mu_i, \sigma_i^2, p_i\}} L(\{x_j\}; \{\mu_i, \sigma_i^2, p_i\})$$

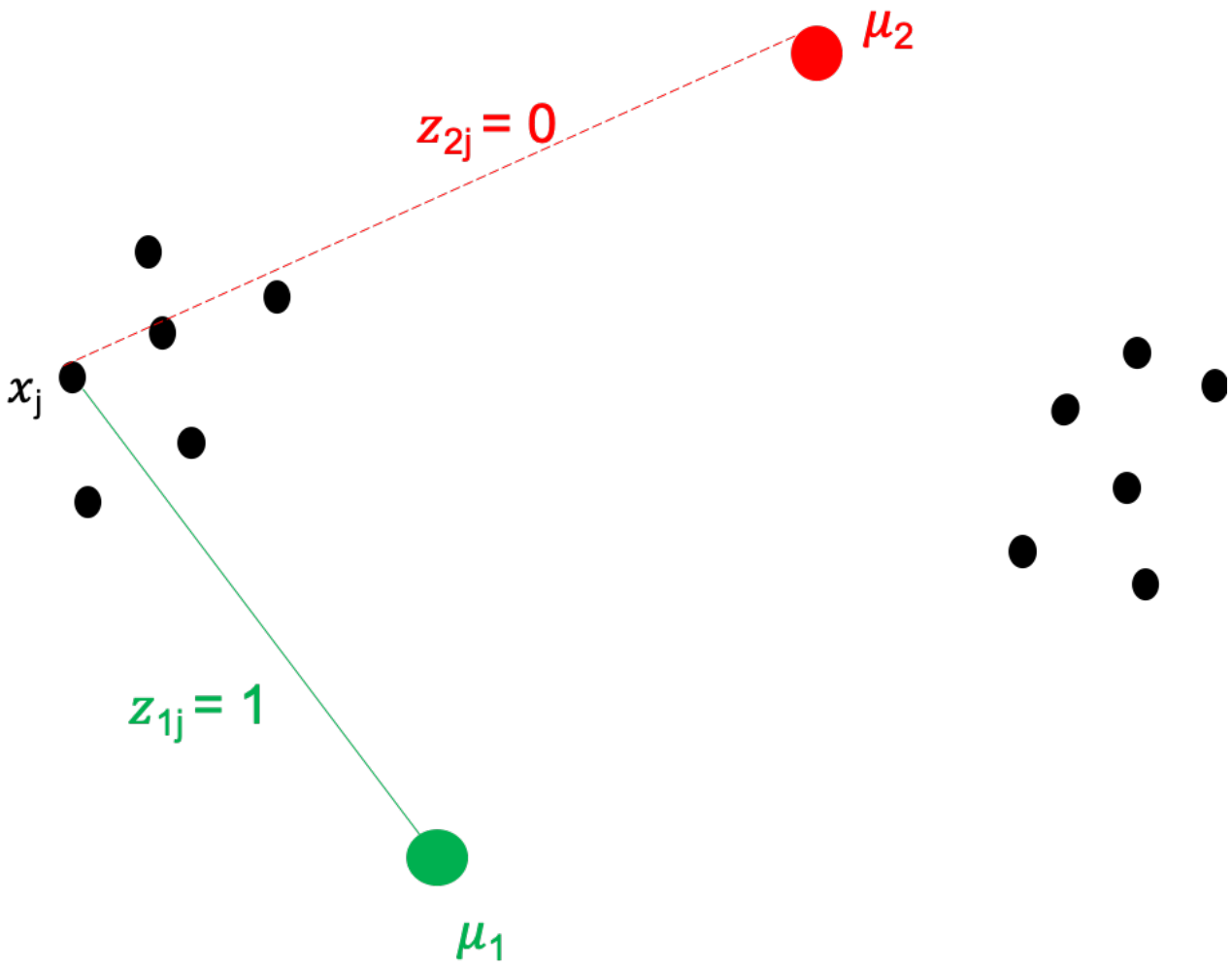
However, this is very complicated (e.g., cannot be made convex by taking log, coupled across all data points etc.)

Idea: Why not use k-means for this ?

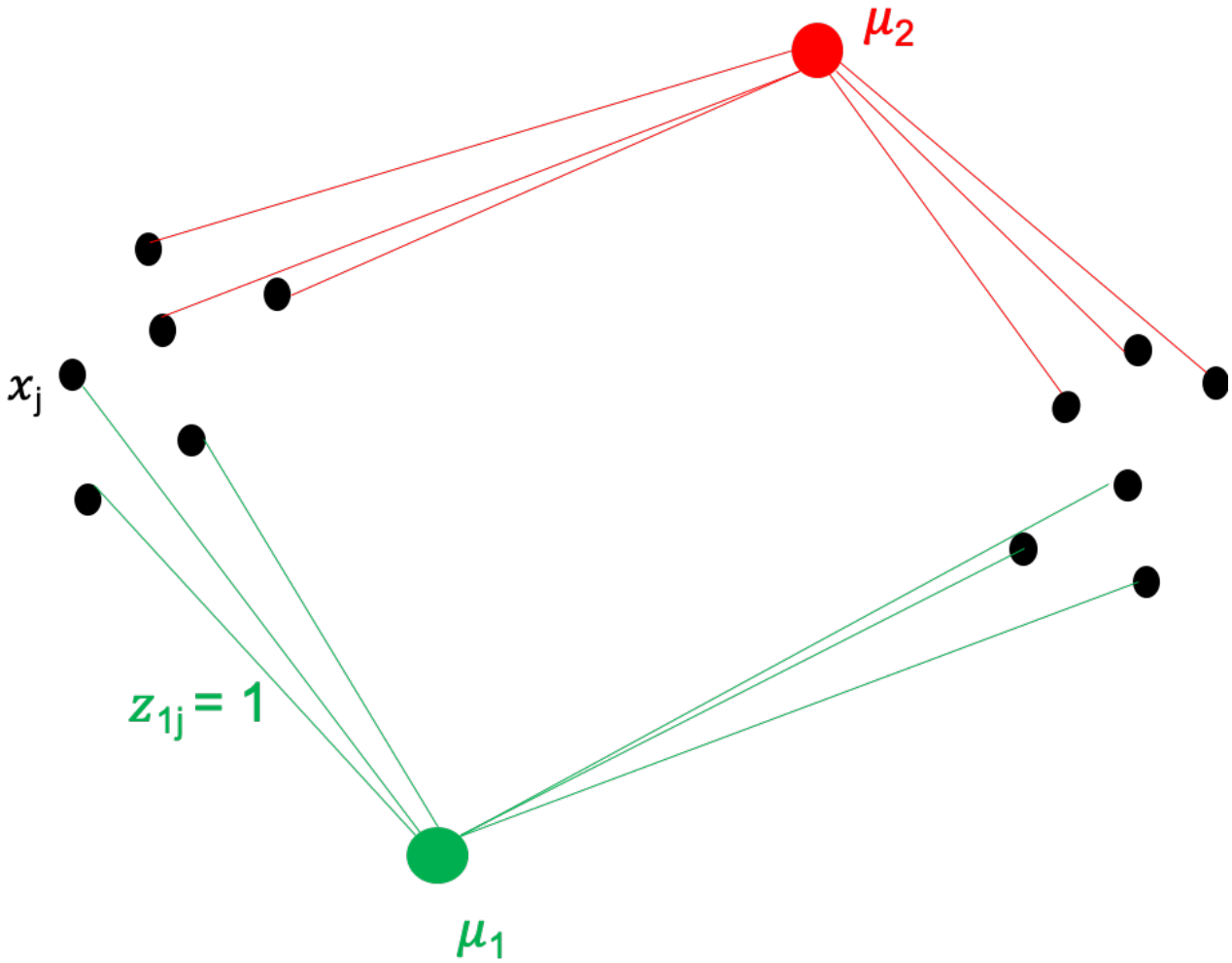
Indicator Variables z_{ij}



Indicator Variables z_{ij}



Indicator Variables z_{ij}



Connecting k-mean and GMMs

Assume: all $\sigma_i = 1$ and all $p_i = 1/K$

$$\tilde{L}(\{x_j\}; \{\mu_i\}\{z_{ij}\}) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-\sum_i (x_j - \mu_i)^2 z_{ij}}{2}\right)$$

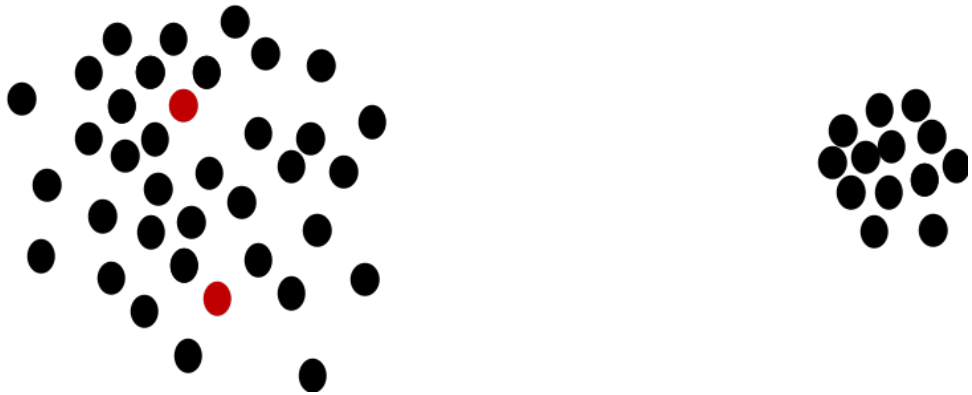
k-means is the following

$$\{\hat{\mu}_i\}_{k-means} = \arg\max_{\{\hat{\mu}_i\}} \left[\max_{\{z_{ij}\}} L\left(\{x_j\}; \{\mu_i\}\{z_{ij}\}\right)\right]$$

where each z_{ij} is 0 or 1 and $\sum_i z_{ij} = 1$

Problems with simple k-means

Cannot account for different co-variances



Cannot account for different p_i, (cluster probabilities i.e. cluster sizes)

Problems with simple k-means

Even if we are very lucky and we

(1) are given that the covariances are all identical and known,

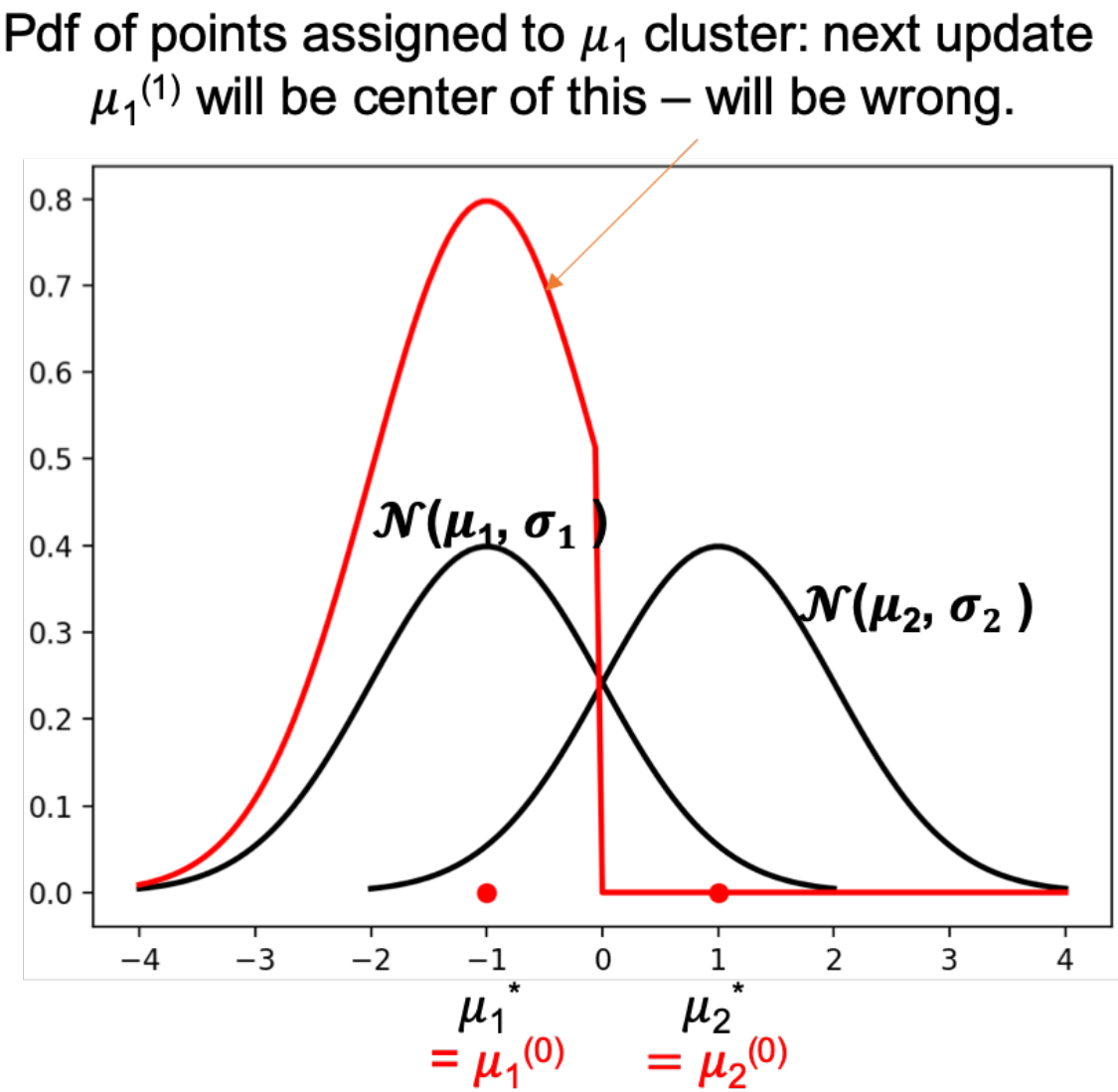
(2) are given that the cluster probabilities are all equal

(3) initialize exactly, i.e. $\mu_i^0 = \mu_i$

(4) even if we have infinite samples

k-means will **still** result in μ_i^t "going away to a wrong place"

Problems with simple k-means



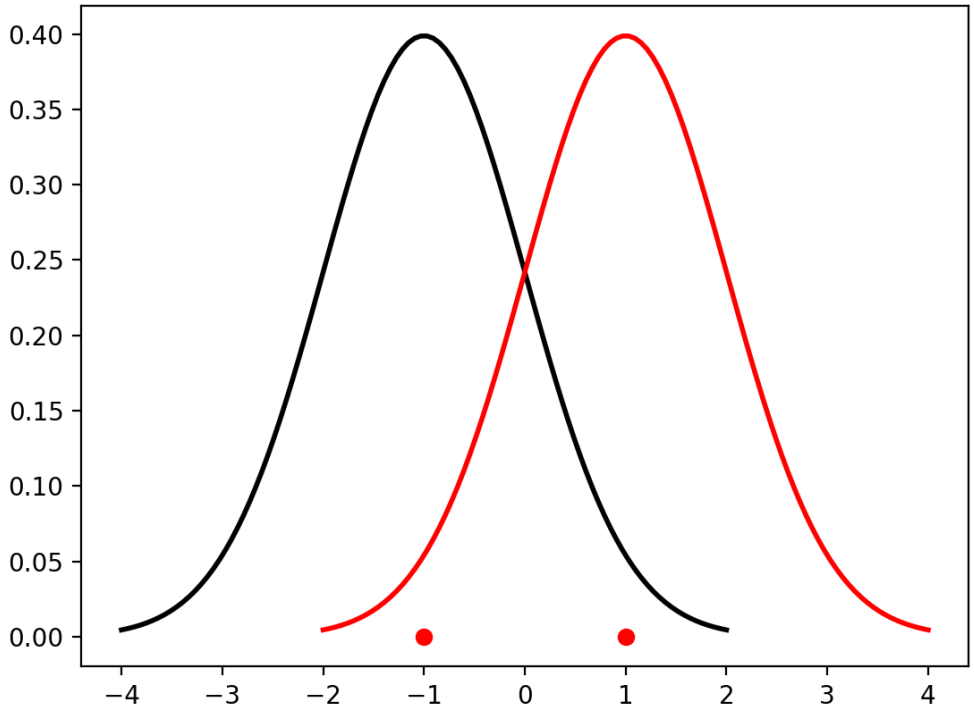
$\mu^{(0)}$'s are the best initialization

Solution to these problems

All of these can be fixed by thinking about how to best solve the GMM problem.

Resulting algo: soft k-means

Soft k-means



k-means

- Each point in exactly one cluster

soft k-means

- Each point j has a probability of being in each cluster

Naive Attempt

k-means objective

$$\min_{\{\mu_i\}} \min_{\{z_{ij}\}} \sum_{i=1}^{k} \sum_{j=1}^{n} ||x_j - \mu_i||^2 z_{ij}$$

$$z_{ij} \in \{0, 1\}, \sum_i z_{ij} = 1 \text{ for all } j$$

so lets try

$$\min_{\{\mu_i\}} \min_{\{w_{ij}\}} \sum_{i=1}^{k} \sum_{j=1}^{n} ||x_j - \mu_i||^2 w_{ij}$$

$$0 \leq w_{ij} \leq 1, \sum_i w_{ij} = 1 \text{ for all } j$$

soft k-means

We should not find the soft assignments w by optimization.

Instead, we should find the **posterior likelihood** of a point x_j being from mixture component $\mathcal{N}(\mu_i, \sigma_i^2)$

Fix a point x_j. Then, for all components i

$$w_{ij} \propto p_i \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{(x_j - \mu_i)^2}{2\sigma_i^2}\right)$$

That is

$$w_{ij} = \frac{p_i \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{(x_j-\mu_i)^2}{2\sigma_i^2}\right)}{\sum_{i'} p_{i'} \frac{1}{\sqrt{2\pi\sigma_{i'}^2}} \exp\left(\frac{(x_j-\mu_{i'})^2}{2\sigma_{i'}^2}\right)}$$

soft k-means

Given the w's, the μ's, σ^2's and p's are updated in the standard way:

$$\mu_i = \frac{\sum_j w_{ij} x_j}{\sum_j w_{ij}}$$

$$p_i = \frac{1}{n} \sum_j w_{ij}$$

$$\sigma_i^2 = \frac{\sum_j w_{ij} (x_j - \mu_i)^2}{\sum_j w_{ij}}$$

Soft k-means involves alternating between updating the w's and updating the $\{\mu, \sigma^2, p\}$ parameters

Now, the **lucky case** is a fixed point of soft k-means.

Soft k-means is more accurate than k-means, but it is slower.