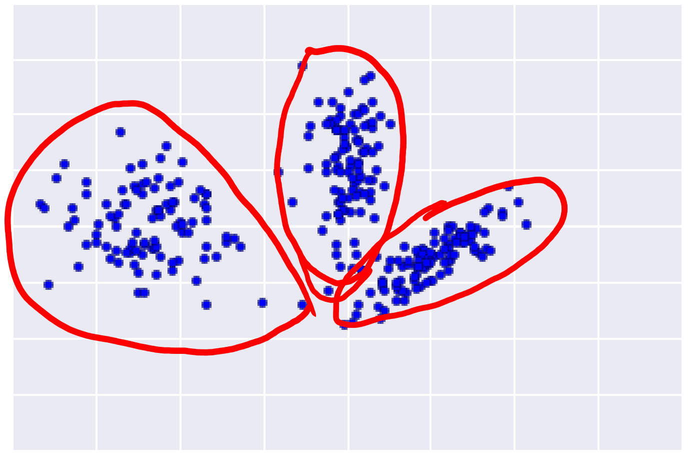
Clustering: K-means and Gaussian Mixture Models

Clustering

- Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$.
- *Clustering* is the following problem: Partition $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ into disjoint subsets called *clusters* such that points in the same cluster are more $\underline{\text{similar}}$ to each other than to points in other clusters.
- A partition of the data points, i.e., a "clustering," can be represented by a function

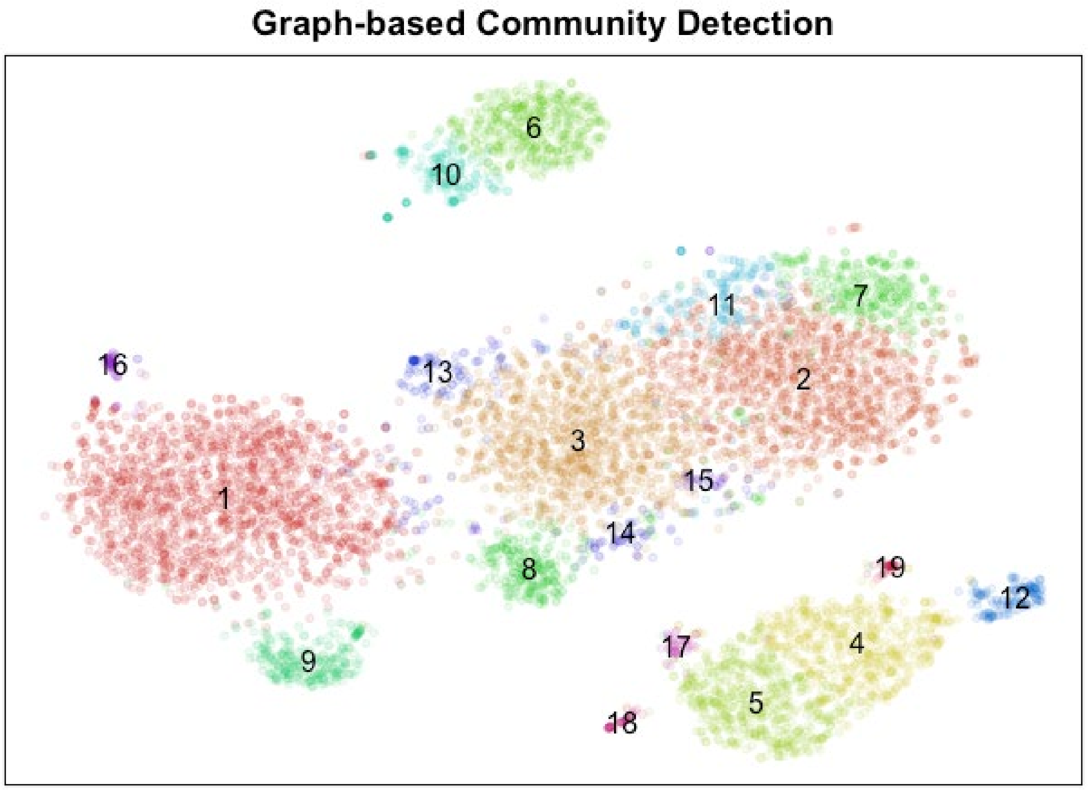
  $$C : \{1, \ldots, n\} \to \{1, \ldots, K\}$$

  where K is the number of clusters. C is called a
- Clustering is unlike supervised learning in that the goal is not to generalize to new data points, but rather to draw conclusions about the training data.
- The definition of clustering above hinges on a notion of similarity



Example: Single Cell Data

- Data matrix $\boldsymbol{X}$. Each column is a feature vector describing a cell.
- Columns of $\boldsymbol{X}$ were clustered using a method called "graph-based community detection"
- Nonlinear dimensionality reduction was performed for visualization



Applications of Clustering

- The main goal of clustering is to discover interesting subpopulations
- Examples:
  - Network traffic analysis: clustering users
  - Marketing/sales: clustering customers
  - Clinical trials: clustering outcomes
  - Finance: cluster stocks
  - Bioinformatics: cluster cells or genes or proteins
- Clustering is often not the end of analysis. Clusters serves to generate hypotheses for subsequent analysis.

K-means

- Suppose there are K total clusters
- Assume $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$
- The idea behind K-means is to represent the ℓ-th cluster by a vector $\boldsymbol{m}_\ell \in \mathbb{R}^d$ called the

  centroid of the ℓth cluster

- If the cluster centroids $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K$ are known, the cluster map is defined to be

$$C(i) = \underset{1 \le \ell \le K}{\arg\min} \; \|x_i - m_\ell\|_2$$

- If the cluster map C is known, the cluster centroids are defined to be

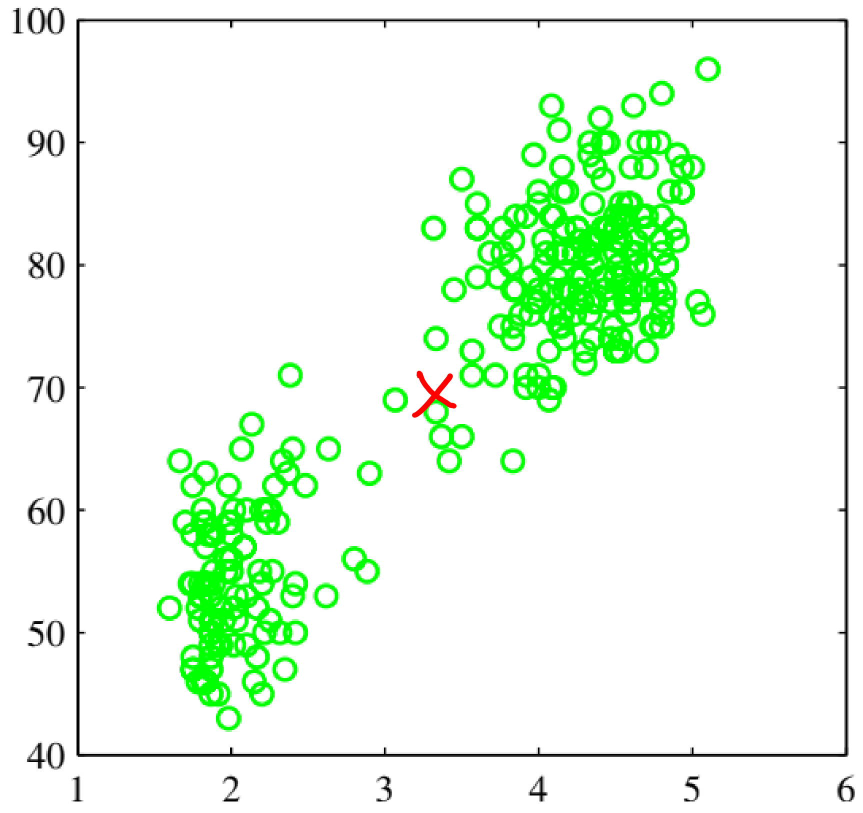
$$m_\ell = \frac{1}{\#\{i : C(i) = \ell\}} \sum_{i : C(i) = \ell} x_i$$

K-means Algorithm

```
Input: $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, $K$
Initialize $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K \in \mathbb{R}^d$
Repeat
    For $i = 1, \ldots, n$
        $C(i) = \arg\min_{\ell} \|\boldsymbol{x}_i - \boldsymbol{m}_\ell\|$
    End
    For $\ell = 1, \ldots, K$
        $\boldsymbol{m}_\ell = \frac{1}{|\{i : C(i) = \ell\}|} \sum_{i:C(i)=\ell} \boldsymbol{x}_i$
    End
Until termination criterion is satisfied
Output: $C$
```

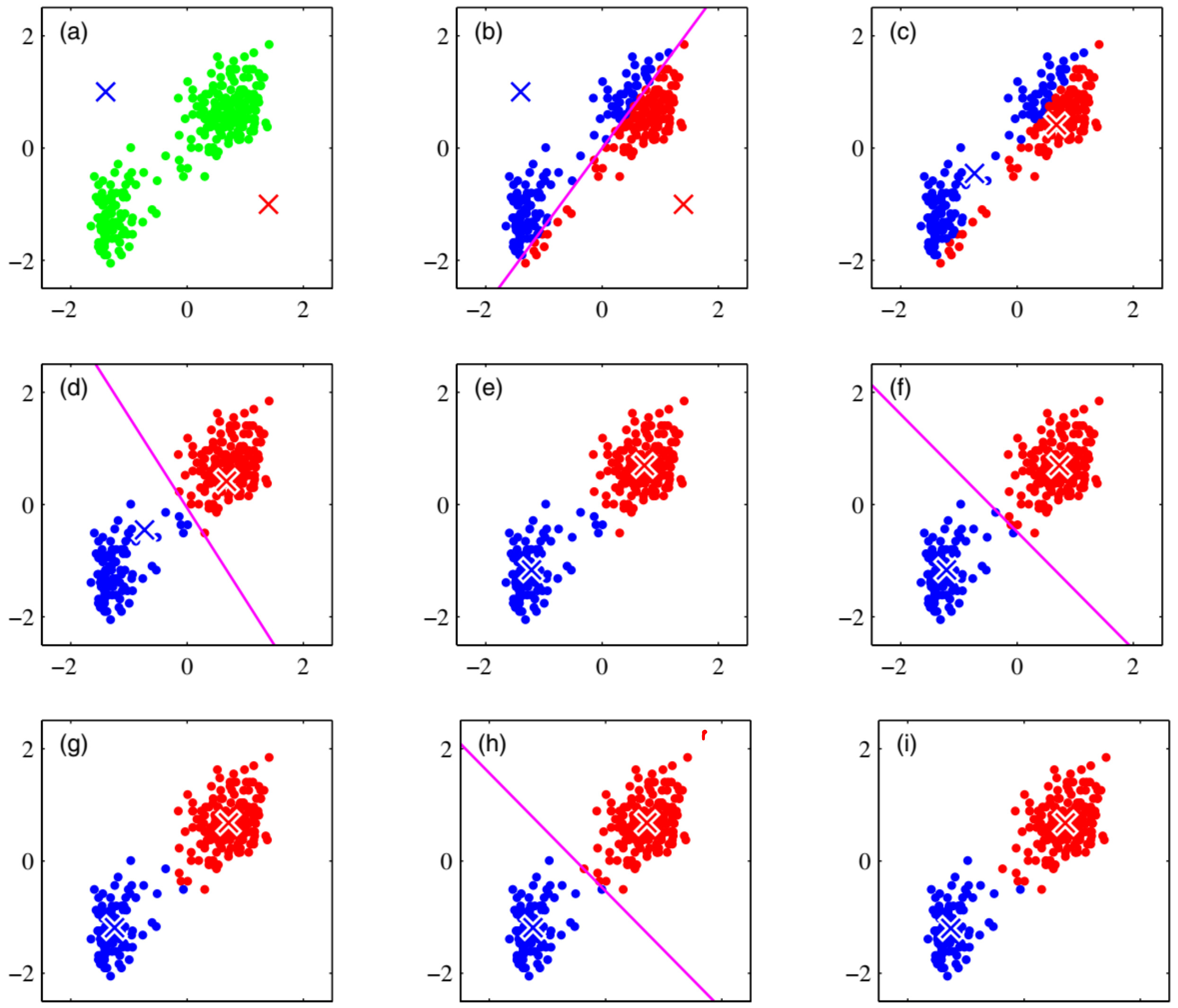
Illustration: Old Faithful Data

Plot of the time to the next eruption in minutes (vertical axis) versus the duration of the eruption in minutes (horizontal axis) for the Old Faithful data set.



Taken from Bishop, *Pattern Recognition and Machine Learning*

Illustration: Old Faithful Data



Data have been rescaled (why?)

Taken from Bishop, *Pattern Recognition and Machine Learning*

K-means Objective

- The K-means algorithm can be viewed as seeking C to minimize

$$W(C) = \sum_{\ell=1}^{k} \frac{1}{n_\ell} \sum_{i:C(i)=\ell} \|x_i - \bar{x}_\ell\|^2$$

  where

$$\bar{\boldsymbol{x}}_\ell = \frac{1}{n_\ell} \sum_{j:C(j)=\ell} \boldsymbol{x}_j, \qquad n_\ell = \#\{i : C(i) = \ell\} = n_\ell(C)$$

- It can be shown that K-means produces a sequence $C_1, C_2, \ldots$ such that

$$W(C_1) \geq W(C_2) \geq \cdots$$

- It can also be shown that

similarity measure is squared Euclidean distance

$$W(C) = \frac{1}{2} \sum_{\ell=1}^{K} \frac{1}{n_\ell} \left[\sum_{i:C(i)=\ell} \sum_{j:C(j)=\ell} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \right].$$

Poll

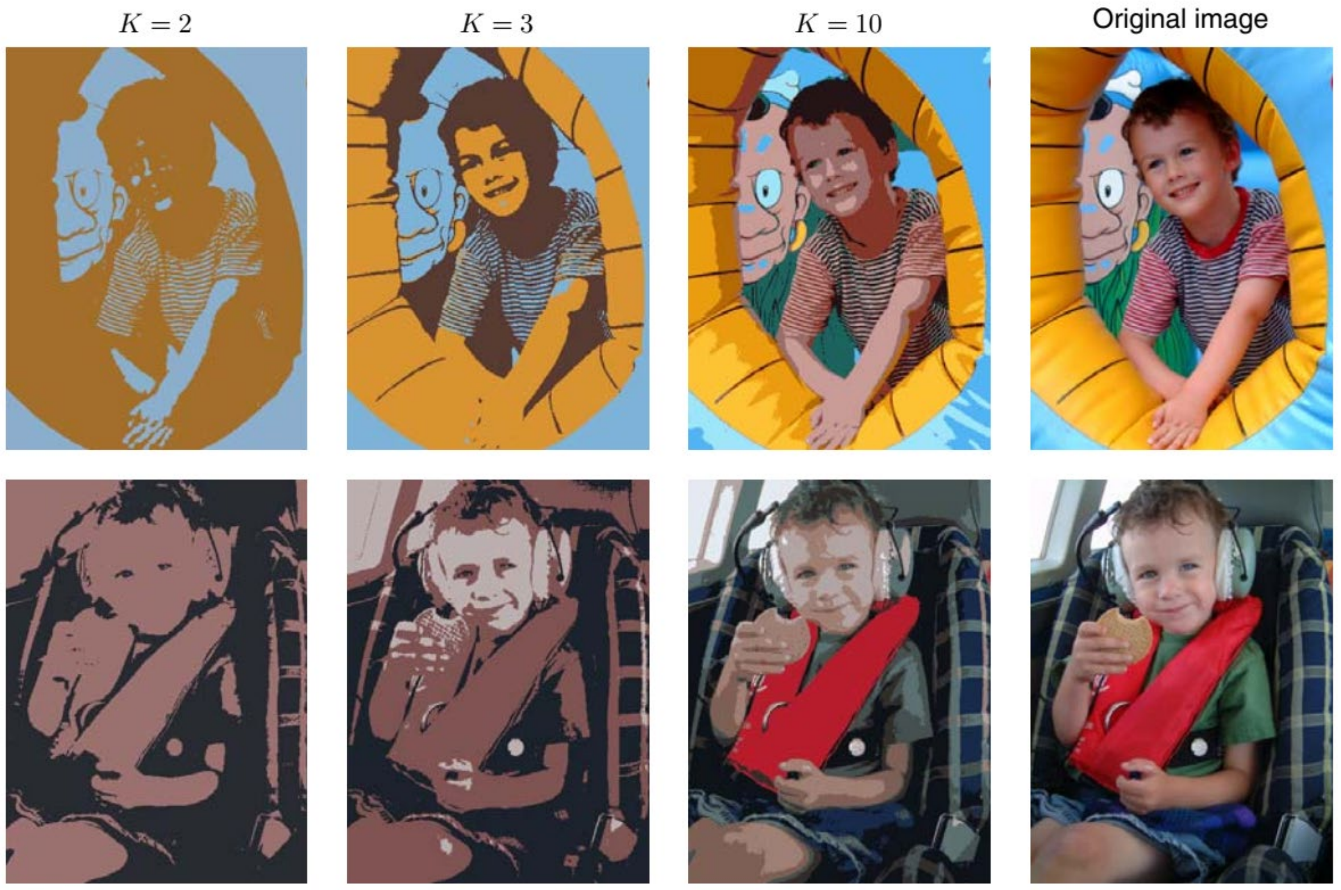
True of False: If we use two different initialzations for two runs of K-means, we can expect the final cluster maps to be the same

(A) True

(B) False

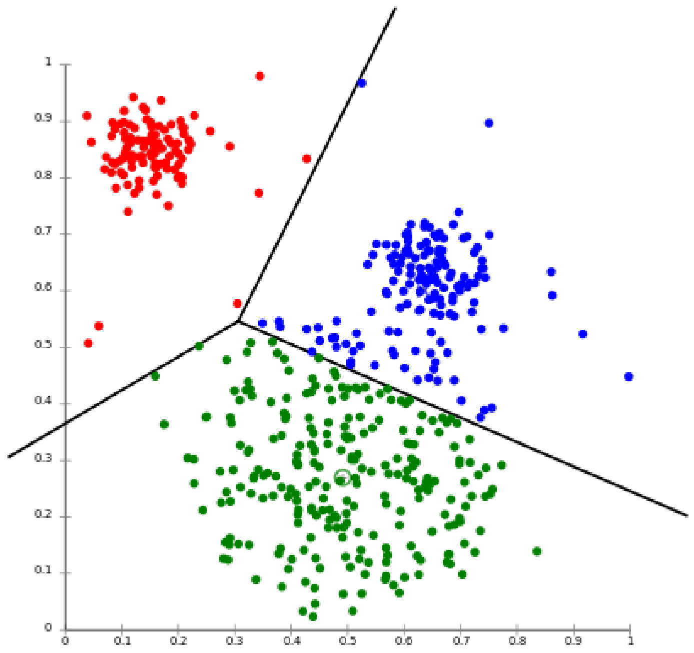
K-means for Image Segmentation

- RGB images, each pixel is a 3-d vector

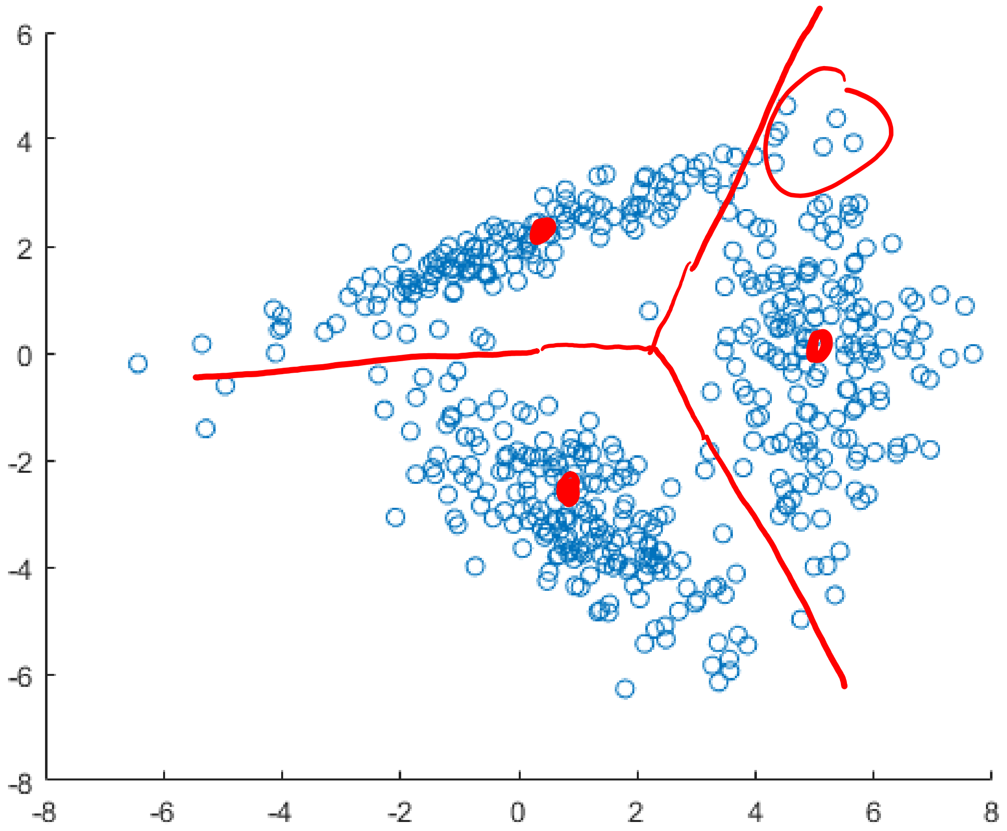


K-means Summary

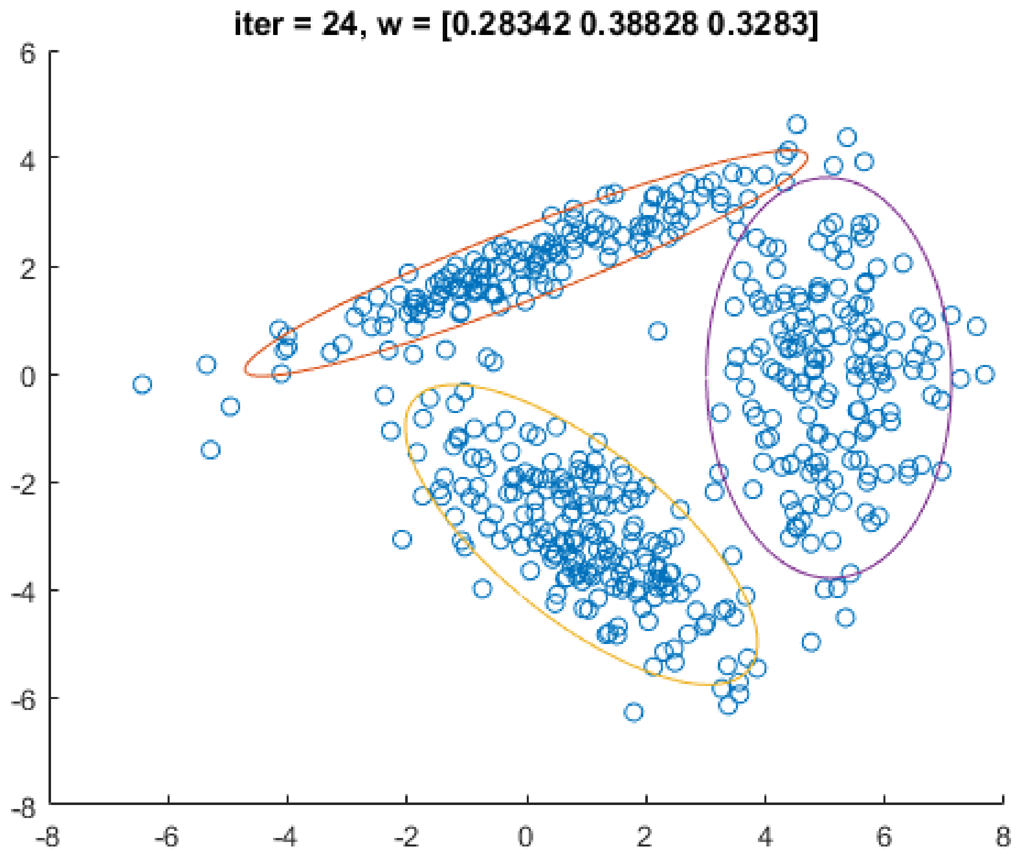
- Underlying notion of similarity: squared Euclidean distance (see second formula for $W(C)$)
- Convex clusters
- Suboptimal convergence



Gaussian Mixture Model



Gaussian Mixture Model



Gaussian Mixture Models

- Let $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma} > 0$.
- Recall the multivariate Gaussian density

$$\phi(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{d}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) \right\}.$$

- A random variable $\boldsymbol{X}$ follows a *Gaussian mixture model* (GMM) *with K components* if its probability density function f has the form

$$f(x) = \sum_{k=1}^{K} w_k \phi(x; \mu_k, \Sigma_k)$$

where $w_k \geq 0$, $\sum_k w_k = 1$, and for all k, $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d\times d}$, $\boldsymbol{\Sigma}_k > 0$.

Poll

True or False: A mixture of two distinct Gaussian densities is another Gaussian density.

(A) True

(B) False

Gaussian Mixture Models

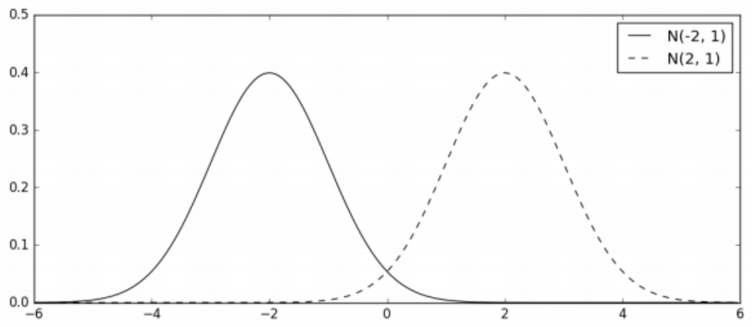


Figure 3: Densities of two different Gaussians RVs.

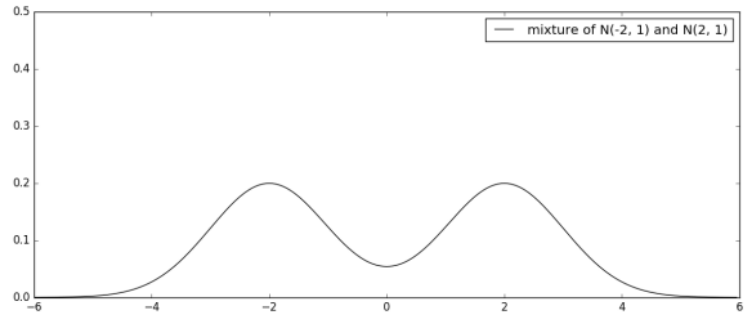


Figure 4: Density of the mixture of the two Gaussians above.

$w_1 = w_2 = \frac{1}{2}$

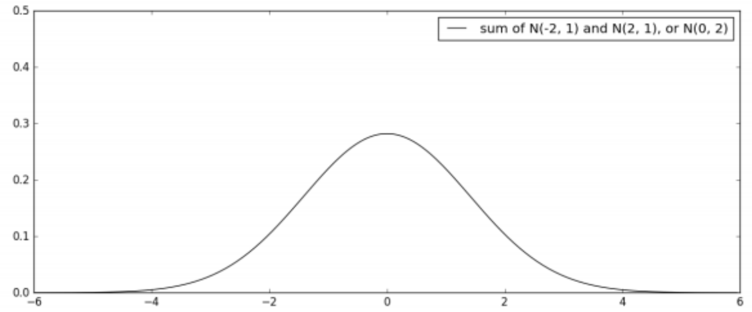


Figure 5: Density of the sum of the two Gaussians above.

Gaussian Mixture Models

- A key to understanding GMMs is to know how to simulate a realization from a known GMM.
- Suppose
$$\boldsymbol{\theta} = (w_1, \ldots, w_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$$
is known.
- The following two-step procedure generates a realization of the GMM with parameter vector $\boldsymbol{\theta}$.
    - First, select $k \in \{1, \ldots, K\}$ at random, according to the pmf $w_1, \ldots, w_K$.
    - Then draw a realization of $\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Maximum Likelihood Estimation

- Observed data: $\underline{\boldsymbol{x}} = \underbrace{(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)}_{\text{independent draws from GMM with param } \theta.}$
- The likelihood is

$$L(\boldsymbol{\theta}; \underline{\boldsymbol{x}}) := \prod_{i=1}^{n} f(\boldsymbol{x}_i; \boldsymbol{\theta})$$

$$= \prod_{i=1}^{n} \left(\sum_{k=1}^{K} w_k \phi(x_i ; \mu_k, \Sigma_k) \right)$$

and the log-likelihood is

$$\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} w_k \phi(x_i ; \mu_k, \Sigma_k) \right)$$

- When $K > 1$, there is no closed form solution for the MLE.

Latent Variables

- We will assume that every observation $\boldsymbol{X}_i$ is associated to an unobserved (or *hidden* or *latent*) variable Z_i $\in \{1, \dots, K\}$

  Z_i is the component from which X_i is drawn

- Let $\underline{\boldsymbol{Z}} = (Z_1, \dots, Z_n)$ denote all the latent variables.
- Note that the random variables $\boldsymbol{X}_i$ and Z_i are jointly distributed
- True or False: The marginal distribution of Z_i is that of a discrete random variable with pmf given by $[w_1, \dots, w_K]$

  (A) True

  (B) False

Alternating Algorithm 1: Hard Cluster Membership

- Simple alternating strategy, similar to K-means
  - Fix $\boldsymbol{\theta}$ and update $z_1, \ldots, z_n$.
  - Fix $z_1, \ldots, z_n$ and update $\boldsymbol{\theta}$

$$z_i = \arg\max_k \ \phi(x_i; \mu_k, \Sigma_k)$$

- The updates are simple:

$$\left\{\begin{aligned} w_k &= \frac{1}{n}\sum_{i=1}^{n} 1_{\{z_i = k\}} \\ \mu_k &= \frac{1}{\#\{i : z_i = k\}} \sum_{i: z_i = k} x_i \\ \Sigma_k &= \frac{1}{\#\{i : z_i = k\}} \sum_{i: z_i = k} (x_i - \mu_k)(x_i - \mu_k)^T \end{aligned}\right.$$

- Limitations of this approach?

Alternating Algorithm ~~1~~ 2: Soft Cluster Membership

- Introduce a notion of soft cluster membership:

$$\gamma_{ik} = \frac{w_k \, \phi(x_i ; \mu_k, \Sigma_k)}{\sum_{\ell=1}^{K} w_\ell \, \phi(x_i ; \mu_\ell, \Sigma_\ell)}$$

- Still a simple alternating strategy
  - Fix $\boldsymbol{\theta}$ and update $\{\gamma_{ik}\}$
  - Fix $\{\gamma_{ik}\}$ and update $\boldsymbol{\theta}$
- The updates are still pretty simple (see next slide)

Algorithm for Learning GMMs

```
Initialize $\boldsymbol{\theta}^{(0)}$, $j = 0$
Repeat
    E-step:
        $$\gamma_{i,k}^{(j)} = \frac{w_k^{(j)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{(j)}, \boldsymbol{\Sigma}_k^{(j)})}{\sum_{\ell=1}^{k} w_\ell^{(j)} \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell^{(j)}, \boldsymbol{\Sigma}_\ell^{(j)})}$$
    M-step:
        $$\boldsymbol{\mu}_k^{(j+1)} = \frac{\sum_i \gamma_{i,k}^{(j)} \boldsymbol{x}_i}{\sum_i \gamma_{i,k}^{(j)}}$$
        $$\boldsymbol{\Sigma}_k^{(j+1)} = \frac{\sum_i \gamma_{i,k}^{(j)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(j+1)})(\boldsymbol{x}_i - \boldsymbol{\mu}_k^{(j+1)})^T}{\sum_i \gamma_{i,k}^{(j)}}$$
        $$w_k^{(j+1)} = \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,k}^{(j)}.$$
    $j = j + 1$
Until convergence criterion satisfied
```

Demo

Connection to K-means

- Consider the GMM

$$f(x) = \sum_{k=1}^{K} w_k \phi(x; \mu_k, \sigma^2 I)$$

where σ^2 is known. The EM algorithm is now to iterate

$$\gamma_{i,k} = \frac{w_k \phi(\boldsymbol{x}_i, \boldsymbol{\mu}_k, \sigma^2 \boldsymbol{I})}{\sum_{\ell=1}^{K} w_\ell \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell, \sigma^2 \boldsymbol{I})}$$

$$\boldsymbol{\mu}_k = \frac{\sum \gamma_{i,k} \boldsymbol{x}_i}{\sum \gamma_{i,k}}$$

$$w_k = \frac{1}{n} \sum_{i=1}^{n} \gamma_{i,k}.$$

As $\sigma^2 \to 0$,

$$\gamma_{i,k} = \begin{cases} 1 & \text{if } k = \arg\min_{\ell} \|x_i - \mu_\ell\| \\ 0 & \text{ow} \end{cases}$$

Next Lecture

- Latent variable models
- The Expectation-Maximization (EM) algorithm

k-means for Image Compression

- Assume N pixels, 8 bits per color
- Total bits to represent uncompressed image

$$24N$$

- To compress the image, replace each pixel by its closest centroid
- Total bits to represent compressed image

$$24k + N\log_2 k$$

k	reduction
2	4.2
3	8.3
4	16.7

- More elaborate schemes are possible. k-means is also known as Lloyd's algorithm or the Lloyd-Max algorithm, and in this context is said to perform *vector quantization*.