TEXAS A&M UNIVERSITY
Engineering

ECEN 758 Data Mining and Analysis: Lecture 9, Expectation Maximization Algorithm

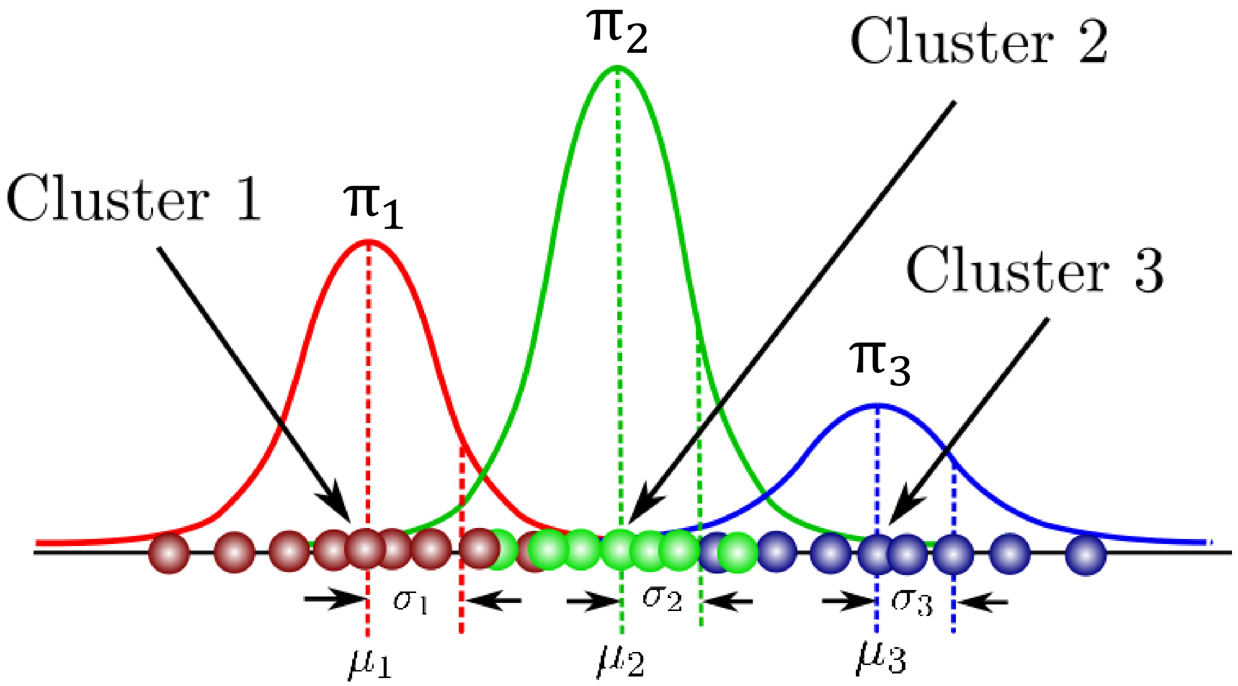
Joshua Peeples, Ph.D.
Assistant Professor
Department of Electrical and Computer Engineering

Announcements

- Assignment #1 grades available
  - Please revise any grade discrepancies within a week (COB, 09/23)
  - Email Dr. Peeples (do not contact Grader) and/or stop by office hours
- Assignment #2 is available now (due 09/27)
  - Please upload submission as single PDF
  - Please share Python code (e.g., Jupyter Notebooks, Google Colab)

Last Lecture

- Gaussian Mixture Models



Gif from: D. Sheehan, Clustering with Scikit with GIFs

Today

- Expectation Maximization Algorithm
- Reading: ZM Chapter 13

Clustering Overview

- We will discuss several variants of clustering
  - **Representative-based Clustering**
  - Hierarchical Clustering
  - Density-Based Clustering

Gaussian Mixture Models Review

Gaussian Mixture Models

- Model clusters as Gaussians
- “Soft” clustering approach
  - Assign probability of belonging to clustering
- Generative model

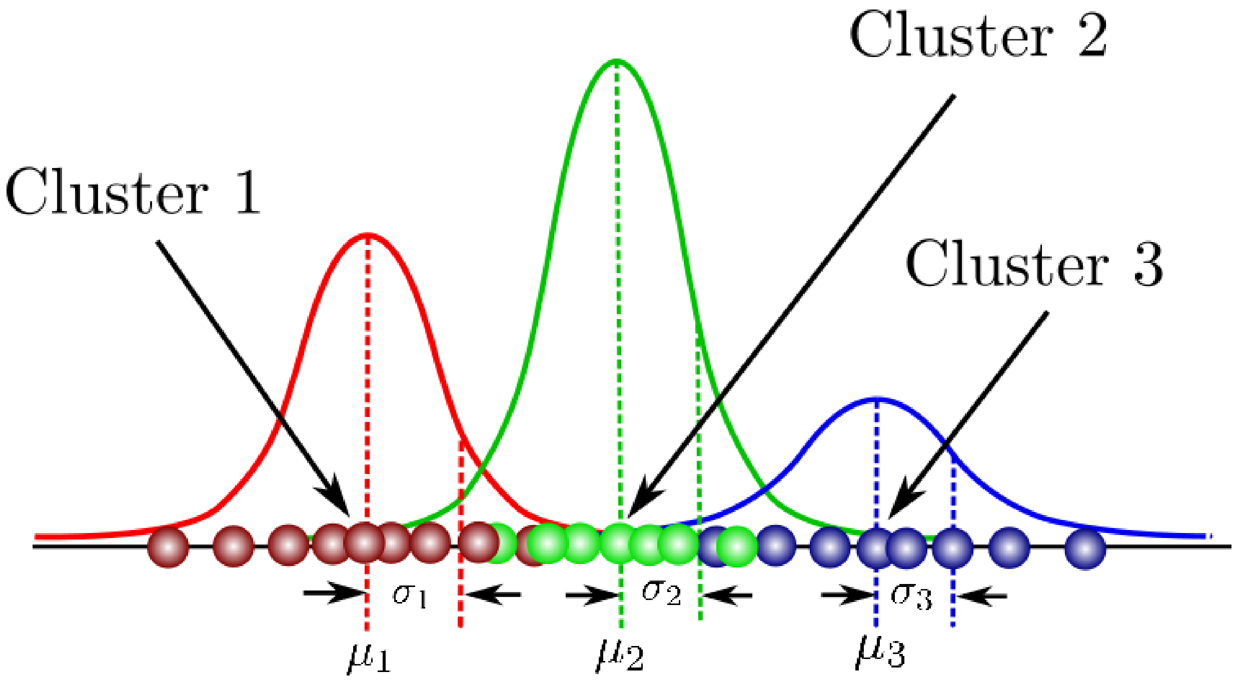
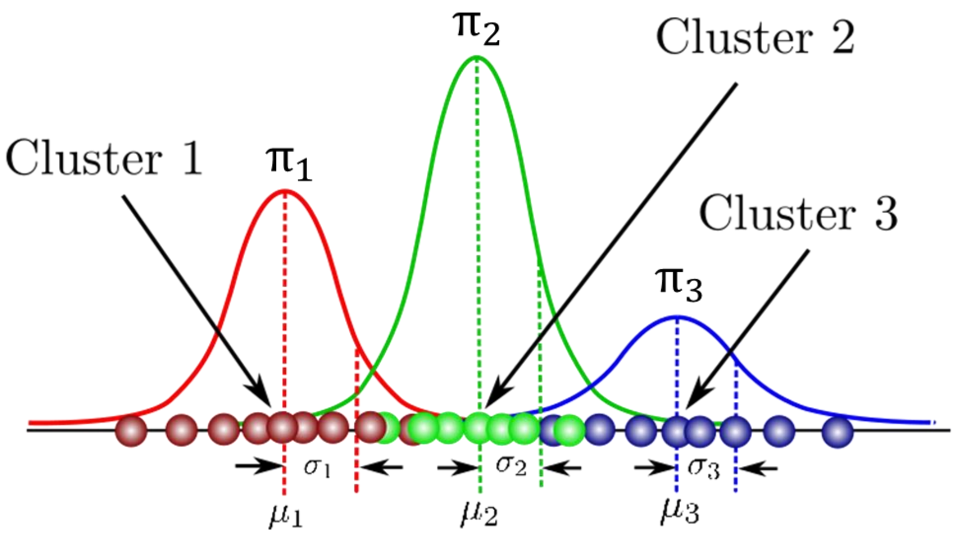


Image from: O. Carrasco, Gaussian Mixture Models Explained.

Mixtures of Gaussians (1D)

- Three parameters to describe clusters:
  - Mean (μ_k)
  - Variance (σ_k^2)
  - Mixture parameters (π_k)
    - Weights, “size”, prior probability
- Probability distribution:

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



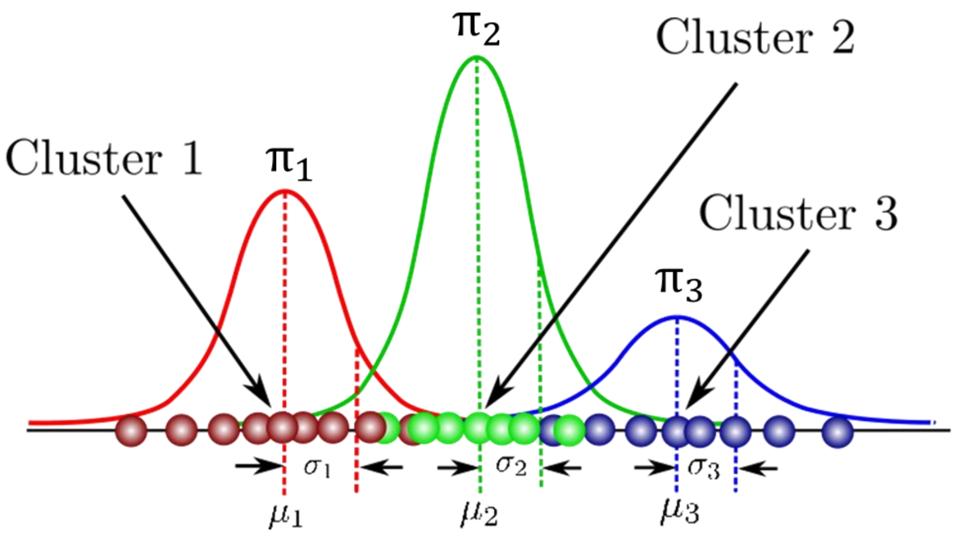
Mixtures of Gaussians (1D)

- Probability distribution:

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

- Select mixture component with probability π_k

$$p(z = k) = \pi_k$$



Slide adapted from: A. Ihler, Gaussian Mixture Models and EM.

Mixtures of Gaussians (1D)

- Probability distribution:

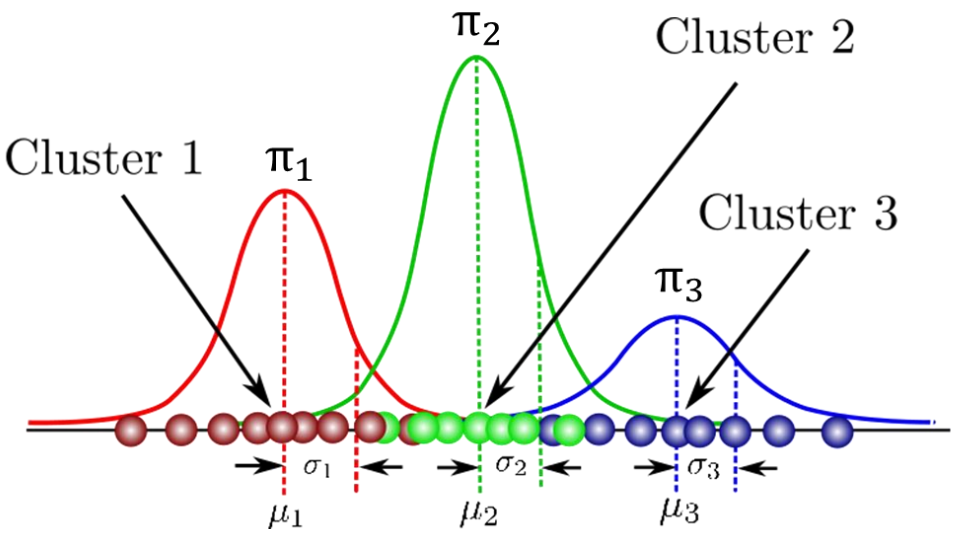
$$p(x) = \sum_{i=1}^{k} \pi_i \mathcal{N}(x|\mu_i, \sigma_i)$$

- Select mixture component with probability π_k

$$p(z = k) = \pi_k$$

- Sample from that component's Gaussian

$$p(x|z = k) = \mathcal{N}(x|\mu_k, \sigma_k)$$



Mixtures of Gaussians (Multivariate)

TEXAS A&M UNIVERSITY Engineering

- Three parameters to describe clusters:
  - Mean vector ($\boldsymbol{\mu}_i$)
  - Covariance matrix (Σ_i^2)
  - Mixture parameters (π_i or $P(C_i)$)
    - Weights, "size", prior probability
    - Sum to one constraint

$$\sum_{i=1}^{k} P(C_i) = 1$$

i^{th} Cluster:

$$f_i(\boldsymbol{x}) = f(\boldsymbol{x}|\boldsymbol{\mu}_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left\{-\frac{(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)}{2}\right\}$$

Probability Density function of **x** as GMM:

$$f(\boldsymbol{x}) = \sum_{i=1}^{k} f_i(\boldsymbol{x})P(C_i) = \sum_{i=1}^{k} f(\boldsymbol{x}|\boldsymbol{\mu}_i, \Sigma_i)P(C_i)$$

Gaussian Mixture Models Algorithm

GMM Algorithm: Objective

- Parameters of model represented as Θ

$$\theta = \{\boldsymbol{\mu}_1, \Sigma_1, P(C_1) \ldots, \boldsymbol{\mu}_k, \Sigma_k, P(C_k)\}$$

- Maximum likelihood estimation (MLE)
- Usually maximize log-likelihood function

Likelihood:

$$P(\mathbf{D}|\boldsymbol{\theta}) = \prod_{j=1}^{n} f(\mathbf{x}_j)$$

MLE:

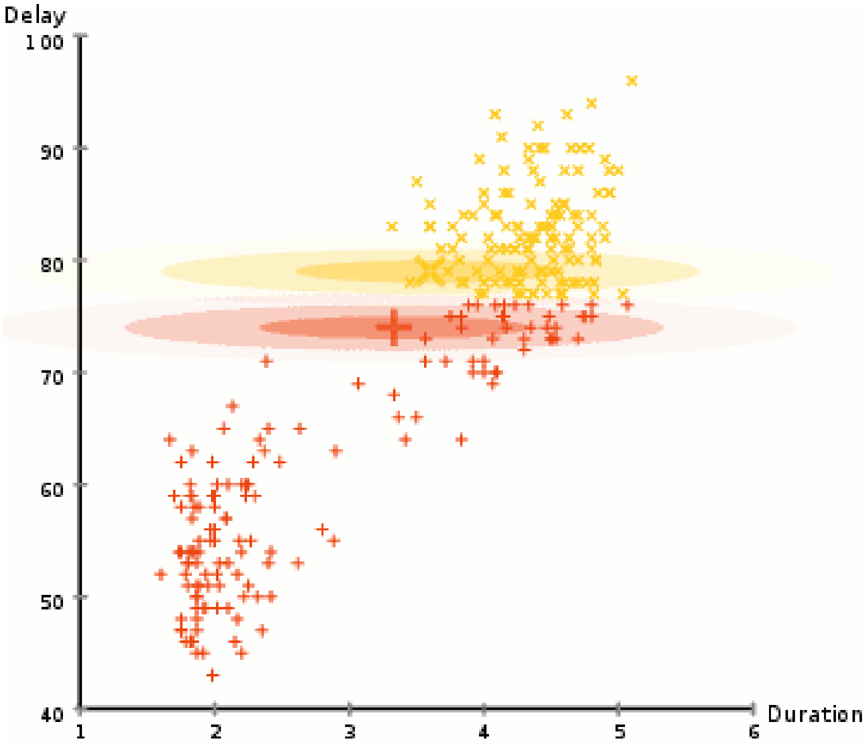
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \{\ln P(\mathbf{D}|\boldsymbol{\theta})\}$$

Log-likelihood:

$$\ln P(\mathbf{D}|\boldsymbol{\theta}) = \sum_{j=1}^{n} \ln f(\mathbf{x}_j) = \sum_{j=1}^{n} \ln \left(\sum_{i=1}^{k} f(\mathbf{x}_j | \boldsymbol{\mu}_i, \Sigma_i) P(C_i) \right)$$

GMM Algorithm: Objective

- Directly maximizing log-likelihood over ϴ is hard
- Alternative approach: Expectation-Maximization (EM)
- Two steps:
  - Expectation: Assignment of points
  - Maximization: Estimation of parameters



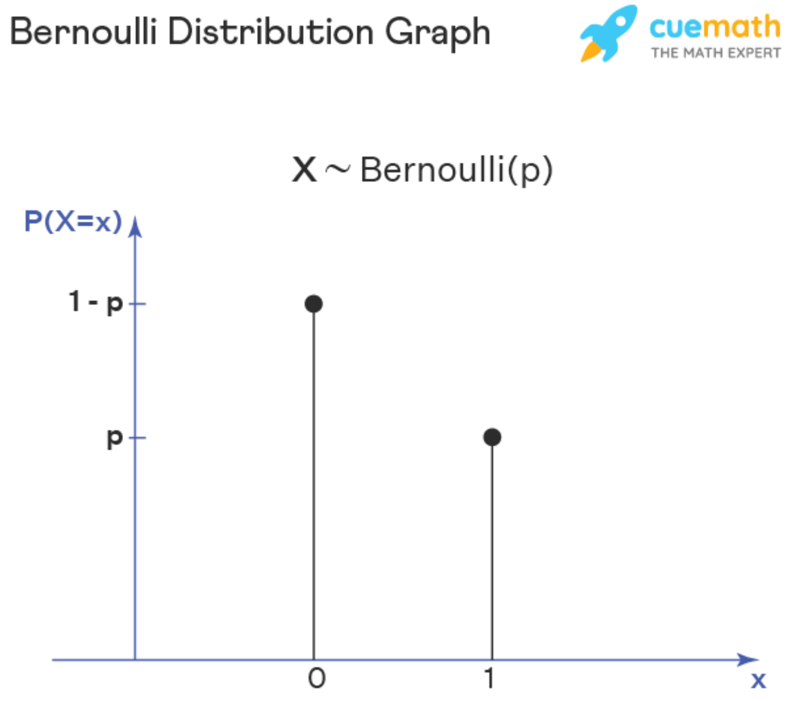
Gif from: Expectation-maximization algorithm, Wikipedia

TEXAS A&M UNIVERSITY Engineering

MLE Example

Probability vs Likelihood

- **Probability:** predict unknown *outcomes* based on known *parameters*
  - $P(\boldsymbol{x} \mid \theta)$
- **Likelihood:** estimate unknown *parameters* based on known *outcomes*:
  - $L(\theta \mid \boldsymbol{x}) = P(\boldsymbol{x} \mid \theta)$
- **Coin-flip example:**
  - θ is probability of "heads" (parameter)
  - x = HHHTTH is outcome from 6 flips
  - Each observation is iid



$$P(X = x) = f(x) = p^x (1 - p)^{1-x}$$

Bernoulli MLE Formulation

- Parameters of model represented as **ϴ**
  - Bernoulli: probability of success, *p*
- Maximum likelihood estimation (MLE)
- Usually maximize log-likelihood function

Likelihood:

$$P(\boldsymbol{D}|\boldsymbol{\theta}) = \prod_{i=1}^{n} f(\boldsymbol{x}_i)$$

$$= \prod_{i=1}^{n} p^{X_i}(1-p)^{1-X_i}$$

$$= p^{\sum_{i=1}^{n} X_i}(1-p)^{n-\sum_{i=1}^{n} X_i}$$

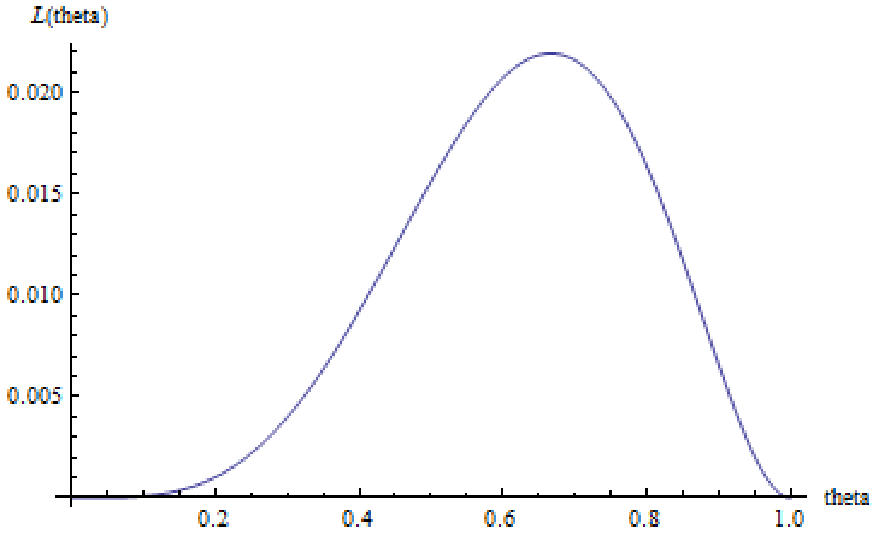
Log-likelihood:

$$\ln P(\boldsymbol{D}|\boldsymbol{\theta}) = \sum_{i=1}^{n} \log p^{X_i}(1-p)^{1-X_i}$$

$$= \sum_{i=1}^{n} X_i(\log p) + (1-X_i) log(1-p)$$

Likelihood for Coin-flip Example

- Probability of outcome given parameter:
  - $P(\mathbf{x} = HHHTTH \mid \theta = 0.5) = 0.5^6 = 0.016$
- Likelihood of parameter given outcome:
  - $L(\theta = 0.5 \mid \boldsymbol{x} = HHHTTH) = P(x \mid \theta) = 0.016$
- Likelihood *maximal* when $\theta = 0.6666$



General Θ:

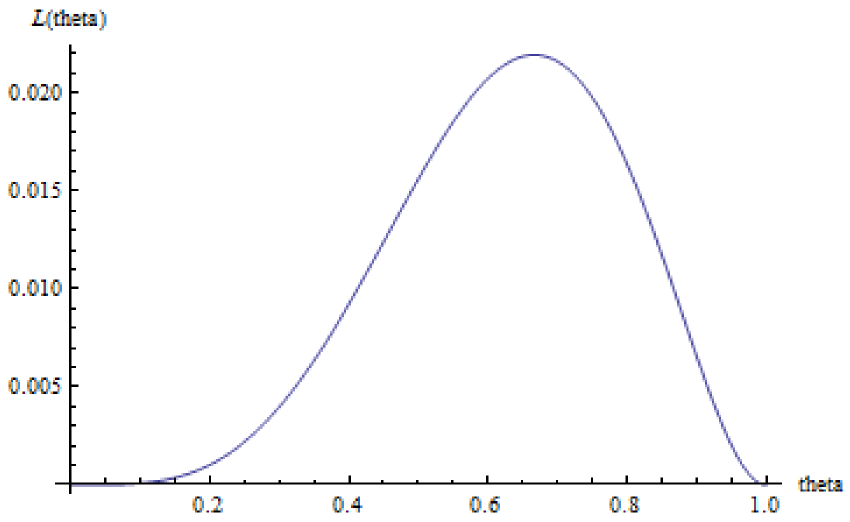
$L(\Theta|HHHTTH) = \Theta^4(1-\Theta)^2$

Coin Flip MLE Details

- $L(\Theta|HHHTTH) = \Theta^4(1-\Theta)^2$
- $\log L(\Theta) = 4 \log \Theta + 2 \log (1-\Theta)$:

  $(d/d\Theta) \log L(\Theta) = 4/\Theta - 2/(1-\Theta)$

  Stationary point: derivative = 0 when $\Theta = 2/3$
- Stationary point is maximizer
  - Because logarithm is a concave function
    - Second derivative is negative
- Intuitive result:
  - MLE of H probability Θ = fraction of H in samples



General Θ:

$L(\Theta|HHHTTH) = \Theta^4(1-\Theta)^2$

Maximum Likelihood Estimation

- Parameterized family of distributions of some r.v. X
- P(**X**|**θ**) for **θ** in some parameter set
- Likelihood L(**θ**,**X**) = P(**X**|**θ**)
- MLE = argmax_θ L(**θ**,**X**)
- Clustering with normal distribution (GMM):
  - Single point $f(x_j) = \Sigma^k_{i=1} f(x_j \mid \mu_i, \Sigma_i)P(C_i)$
  - $P[X|\theta]=\text{Prod}_j\ f(x_j)$
  - Log-LLHD
  - $\log P(X|\theta)=\Sigma^n_{j=1} \log f(x_j) = \Sigma^n_{j=1} \log \Sigma^k_{i=1} f(x_j \mid \mu_i, \Sigma_i)P(C_i)$
- Find max by differentiation?
  - Difficult due to sum inside logarithms

Likelihood:

$$P(\mathbf{D}|\theta) = \prod_{j=1}^{n} f(\mathbf{x}_j)$$

MLE:

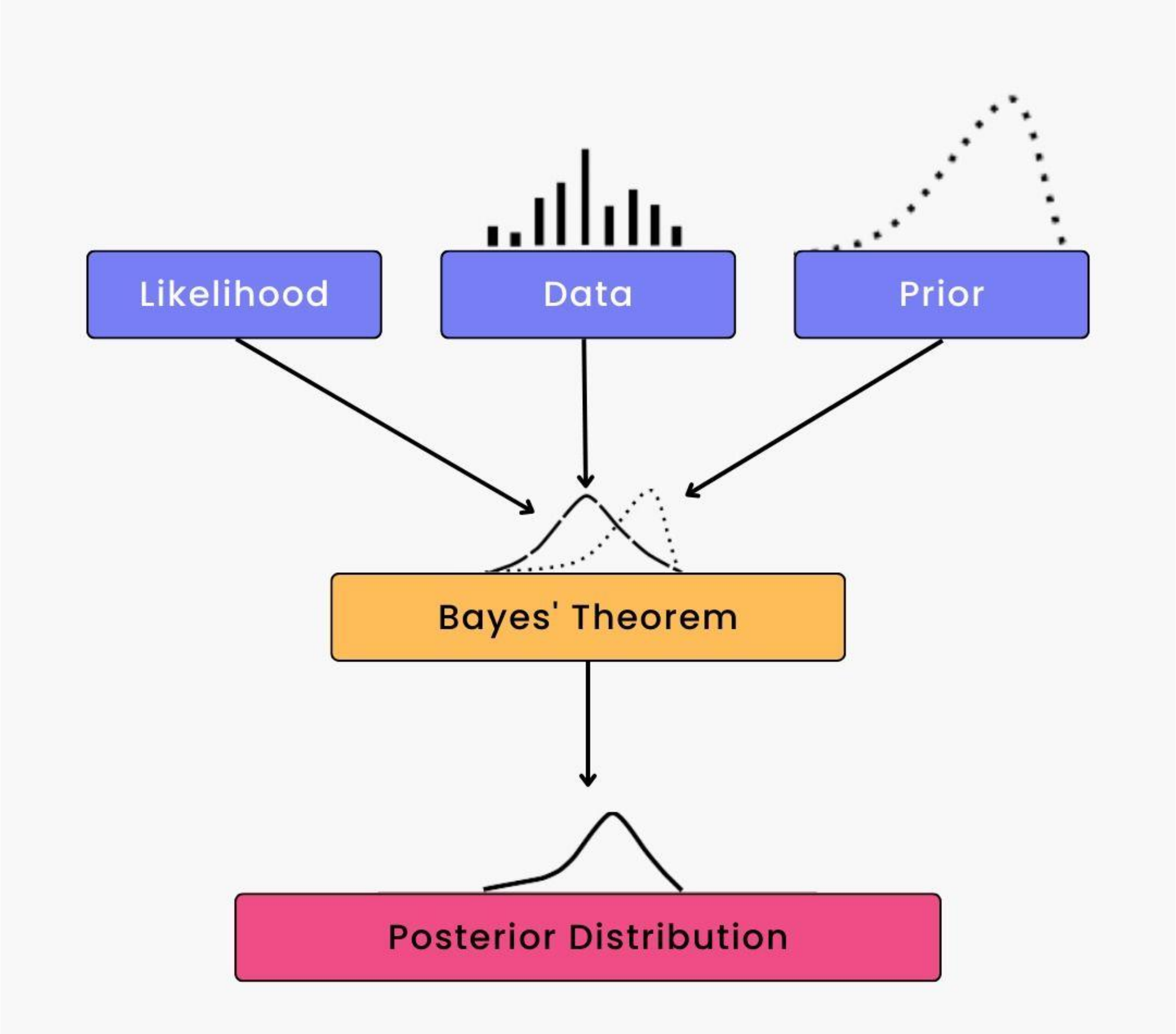
$$\theta^* = \arg\max_{\theta}\{\ln P(\mathbf{D}|\theta)\}$$

Log-likelihood:

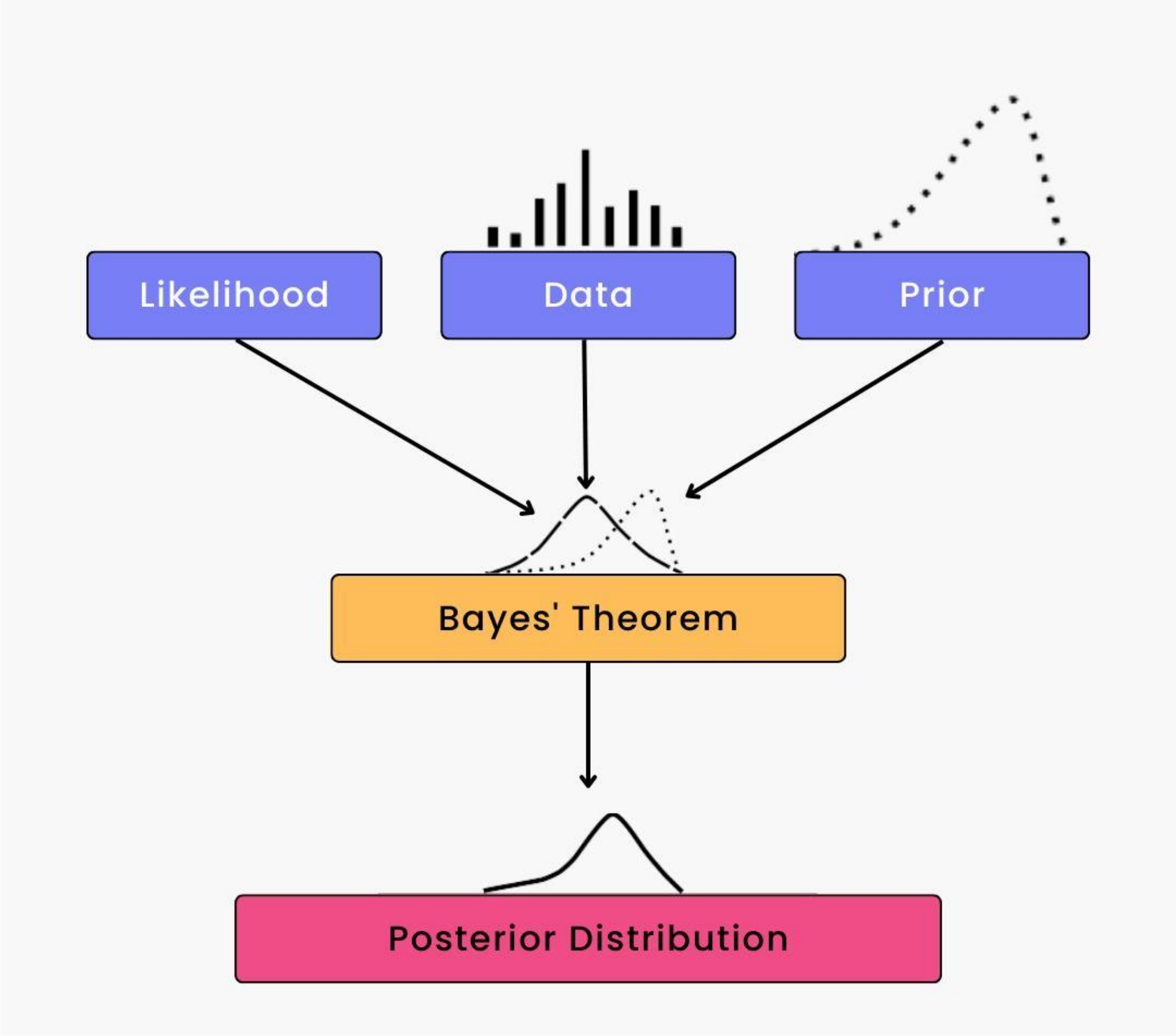
$$\ln P(\mathbf{D}|\theta) = \sum_{j=1}^{n} \ln f(\mathbf{x}_j) = \sum_{j=1}^{n} \ln\left(\sum_{i=1}^{k} f(\mathbf{x}_j|\boldsymbol{\mu}_i, \Sigma_i)P(C_i)\right)$$

Bayes' Theorem

Bayes' Theorem



Bayes' Theorem



$$\underbrace{P(y|x)}_{\text{"Posterior"}} = \frac{\overbrace{P(x|y)}^{\text{"Likelihood"}}}{\underbrace{P(x)}_{\text{"Evidence"}}} \overbrace{P(y)}^{\text{"Prior"}}$$

EM Algorithm and Bayes' Theorem

- Use Bayes' theorem to compute cluster posterior probabilities
- Use posterior probabilities to estimate parameters of model

$$\underbrace{P(y|x)}_{\text{"Posterior"}} = \frac{\overbrace{P(x|y)}^{\text{"Likelihood"}}}{\underbrace{P(x)}_{\text{"Evidence"}}} \overbrace{P(y)}^{\text{"Prior"}}$$

$$P(C_i|\mathbf{x}_j) = \frac{P(C_i \text{ and } \mathbf{x}_j)}{P(\mathbf{x}_j)} = \frac{P(\mathbf{x}_j|C_i)P(C_i)}{\sum_{a=1}^{k} P(\mathbf{x}_j|C_a)P(C_a)} = \frac{f_i(\mathbf{x}_j) \cdot P(C_i)}{\sum_{a=1}^{k} f_a(\mathbf{x}_j) \cdot P(C_a)}$$

TEXAS A&M UNIVERSITY Engineering

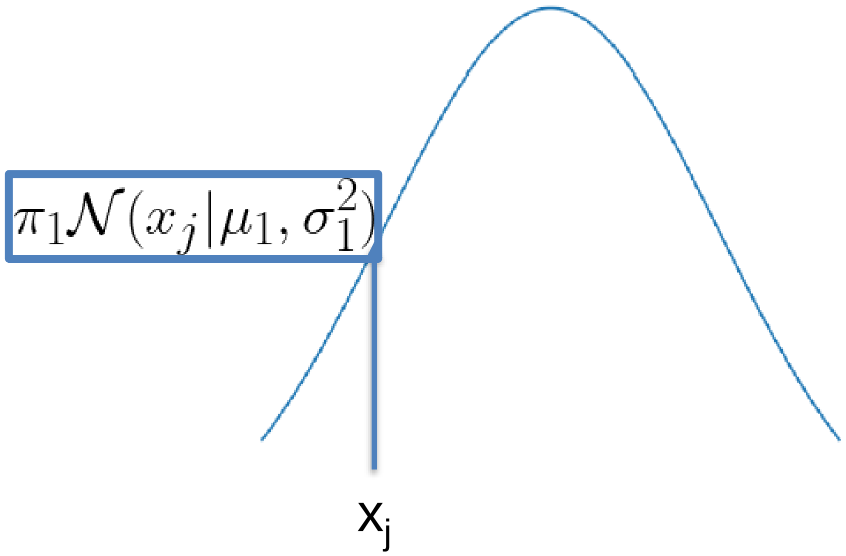
GMM Expectation-Maximization (1D)

GMM Expectation-Maximization (1D): E-step

- **Initialize cluster parameters**
- Expectation (E-Step)
  - For each data point, x_j
  - Compute cluster posterior probability
    - Compute probability with respect to C_i
    - Normalize to sum to one over clusters

For each cluster:

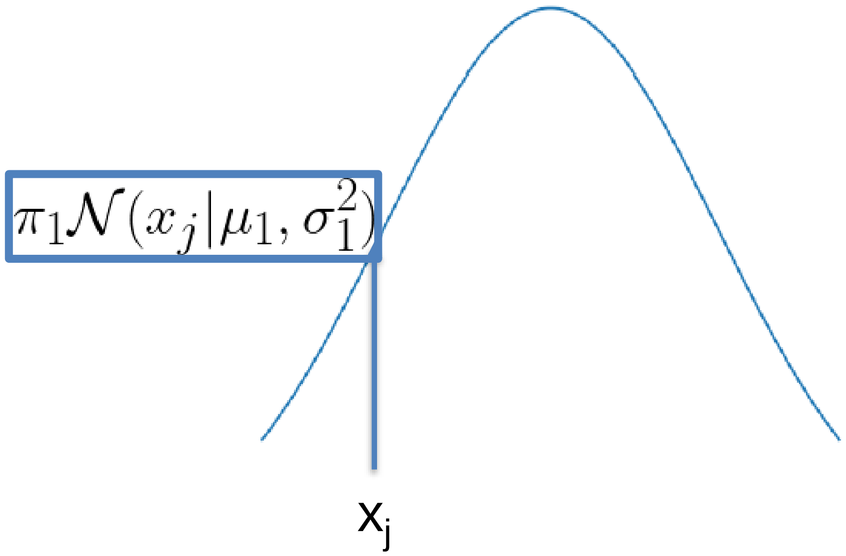
$$f_i(x) = f(x|\mu_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\}$$



GMM Expectation-Maximization (1D): E-step

- Initialize cluster parameters
- Expectation (E-Step)
  - For each data point, x_j
  - **Compute cluster posterior probability**
    - Compute probability with respect to C_i
    - Normalize to sum to one over clusters

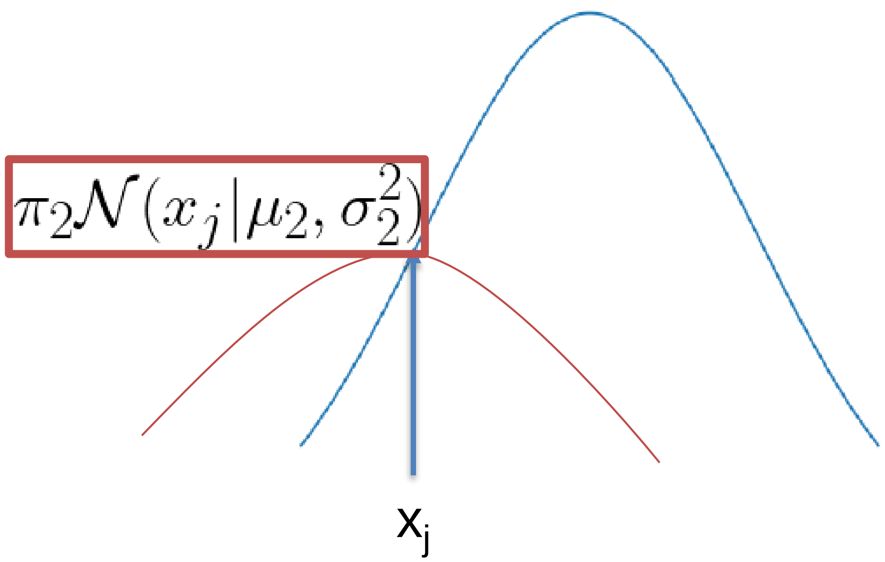
$$w_{ij} = P(C_i|x_j) = \frac{f(x_j|\mu_i, \sigma_i^2) \cdot P(C_i)}{\sum_{a=1}^{k} f(x_j|\mu_a, \sigma_a^2) \cdot P(C_a)}$$



GMM Expectation-Maximization (1D): E-step

- Expectation (E-Step)
  - For each data point, x_j
  - **Compute cluster posterior probability**
    - Compute probability with respect to C_i
    - Normalize to sum to one over clusters

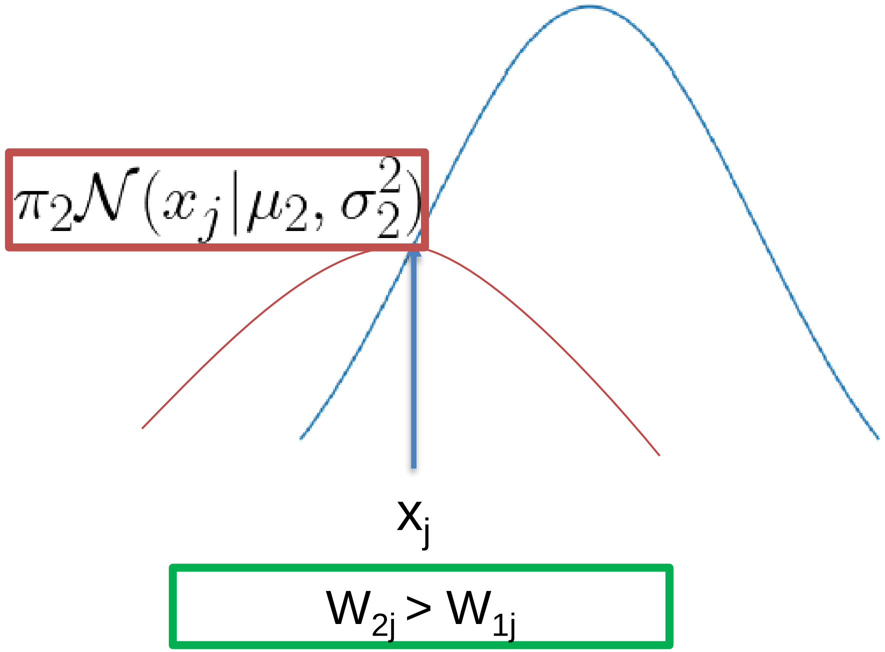
$$w_{ij} = P(C_i|x_j) = \frac{f(x_j|\mu_i, \sigma_i^2) \cdot P(C_i)}{\sum_{a=1}^{k} f(x_j|\mu_a, \sigma_a^2) \cdot P(C_a)}$$



GMM Expectation-Maximization (1D): E-step

- Expectation (E-Step)
  - For each data point, x_j
  - Compute cluster posterior probability
    - Compute probability with respect to C_i
    - Normalize to sum to one over clusters
- **Higher probability will be assigned to Gaussian that is more likely**

$$w_{ij} = P(C_i|x_j) = \frac{f(x_j|\mu_i,\sigma_i^2)\cdot P(C_i)}{\sum_{a=1}^{k} f(x_j|\mu_a,\sigma_a^2)\cdot P(C_a)}$$



GMM Expectation-Maximization (1D): M-step

- Maximization (M-Step)
  - Update parameters using (weighted) data points

$$w_{ij} = P(C_i|x_j) = \frac{f(x_j|\mu_i,\sigma_i^2)\cdot P(C_i)}{\sum_{a=1}^{k} f(x_j|\mu_a,\sigma_a^2)\cdot P(C_a)}$$

Mean:

$$\mu_i = \frac{\sum_{j=1}^{n} w_{ij}\cdot x_j}{\sum_{j=1}^{n} w_{ij}}$$

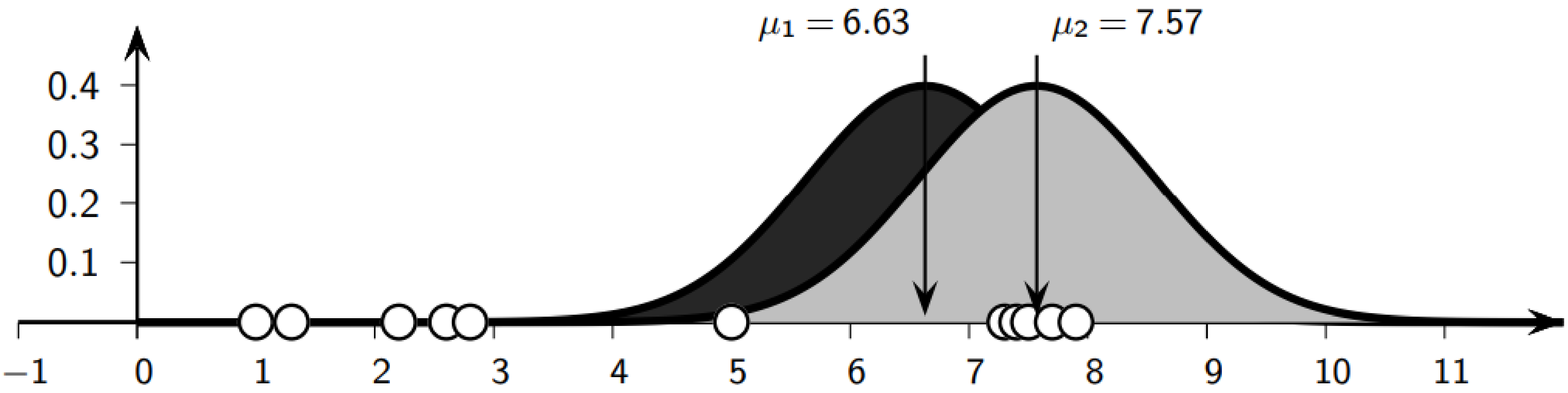
Variance:

$$\sigma_i^2 = \frac{\sum_{j=1}^{n} w_{ij}(x_j-\mu_i)^2}{\sum_{j=1}^{n} w_{ij}}$$

Mixture Weight/Prior Probability:

$$P(C_i) = \frac{\sum_{j=1}^{n} w_{ij}}{n}$$

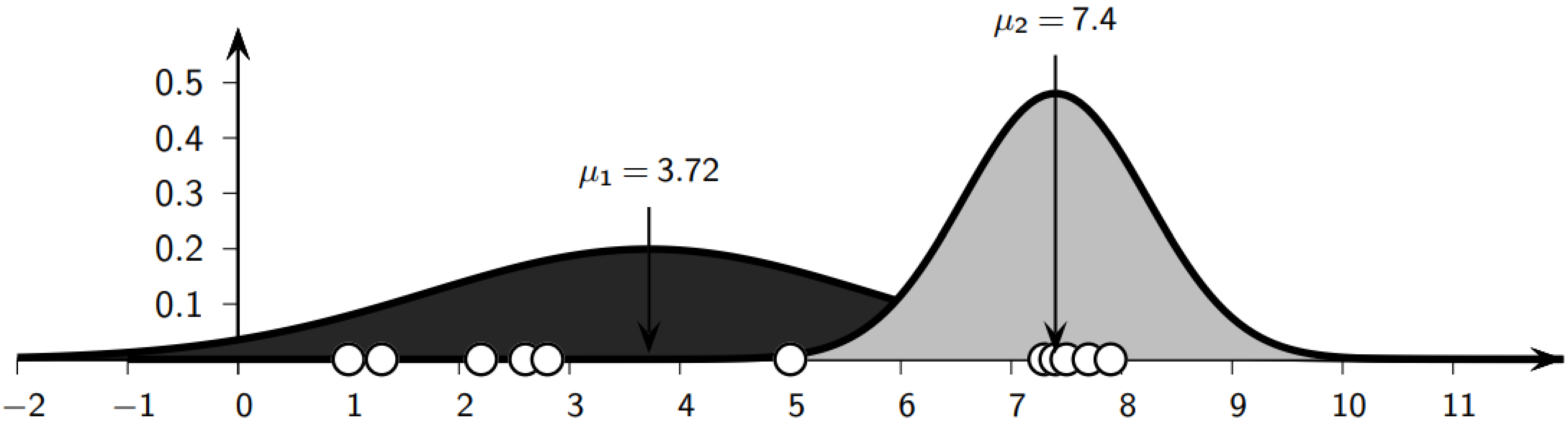
GMM EM 1D Example



(a) Initialization: $t = 0$

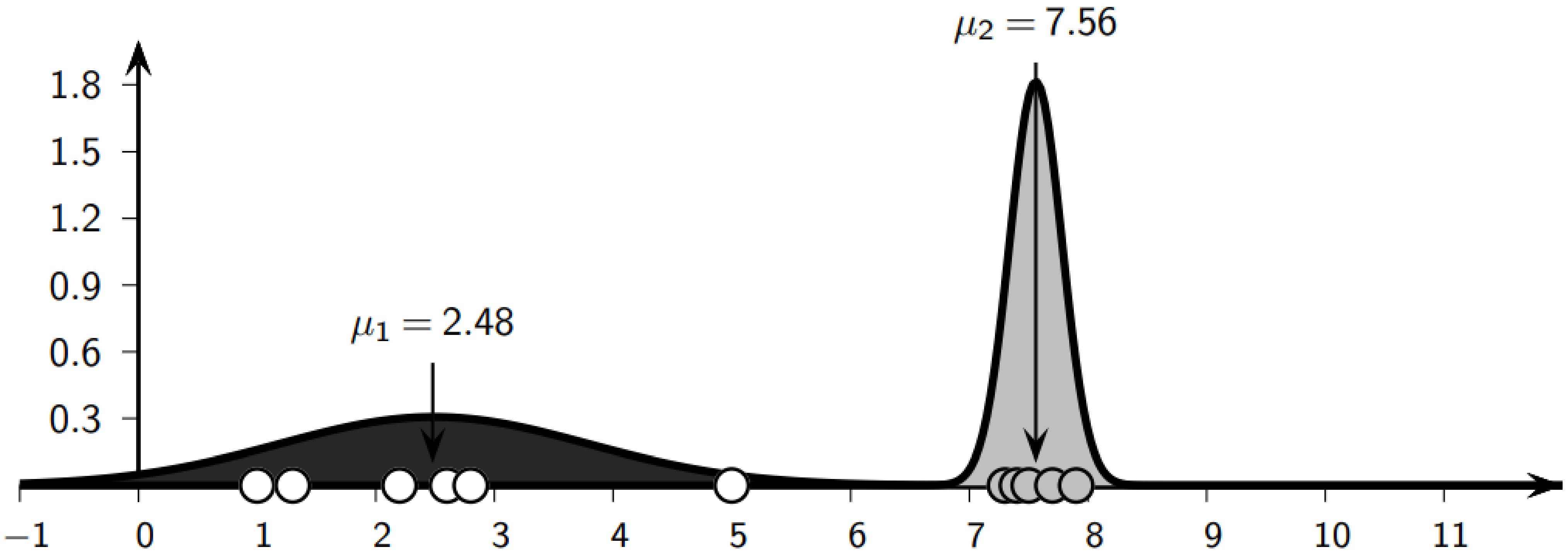
GMM EM 1D Example

TEXAS A&M UNIVERSITY Engineering



(b) Iteration: $t = 1$

GMM EM 1D Example



(c) Iteration: $t = 5$ (converged)

GMM Expectation-Maximization (d-dimensions)

EM in d Dimensions

- Each cluster will have d x d covariance matrix
- Expensive to calculate and may be unreliable estimation
- Can use diagonal covariance
  - Assumes dimensions are independent

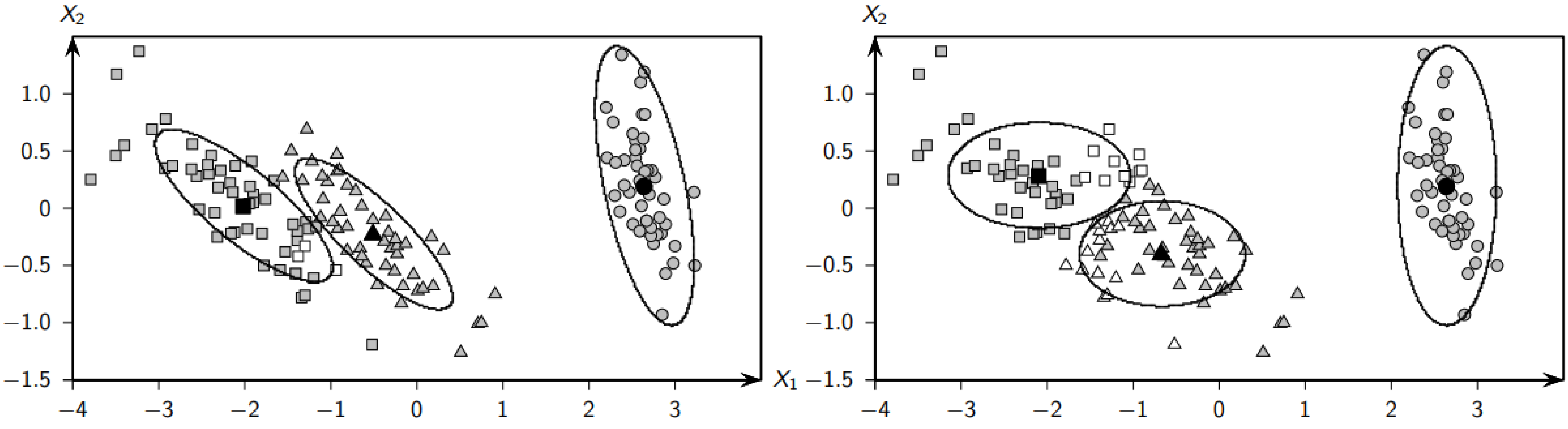
Full Covariance:

$$\Sigma_i = \begin{pmatrix} (\sigma_1^i)^2 & \sigma_{12}^i & \cdots & \sigma_{1d}^i \\ \sigma_{21}^i & (\sigma_2^i)^2 & \cdots & \sigma_{2d}^i \\ \vdots & \vdots & \ddots & \\ \sigma_{d1}^i & \sigma_{d2}^i & \cdots & (\sigma_d^i)^2 \end{pmatrix}$$

Diagonal Covariance:

$$\Sigma_i = \begin{pmatrix} (\sigma_1^i)^2 & 0 & \cdots & 0 \\ 0 & (\sigma_2^i)^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ 0 & 0 & \cdots & (\sigma_d^i)^2 \end{pmatrix}$$

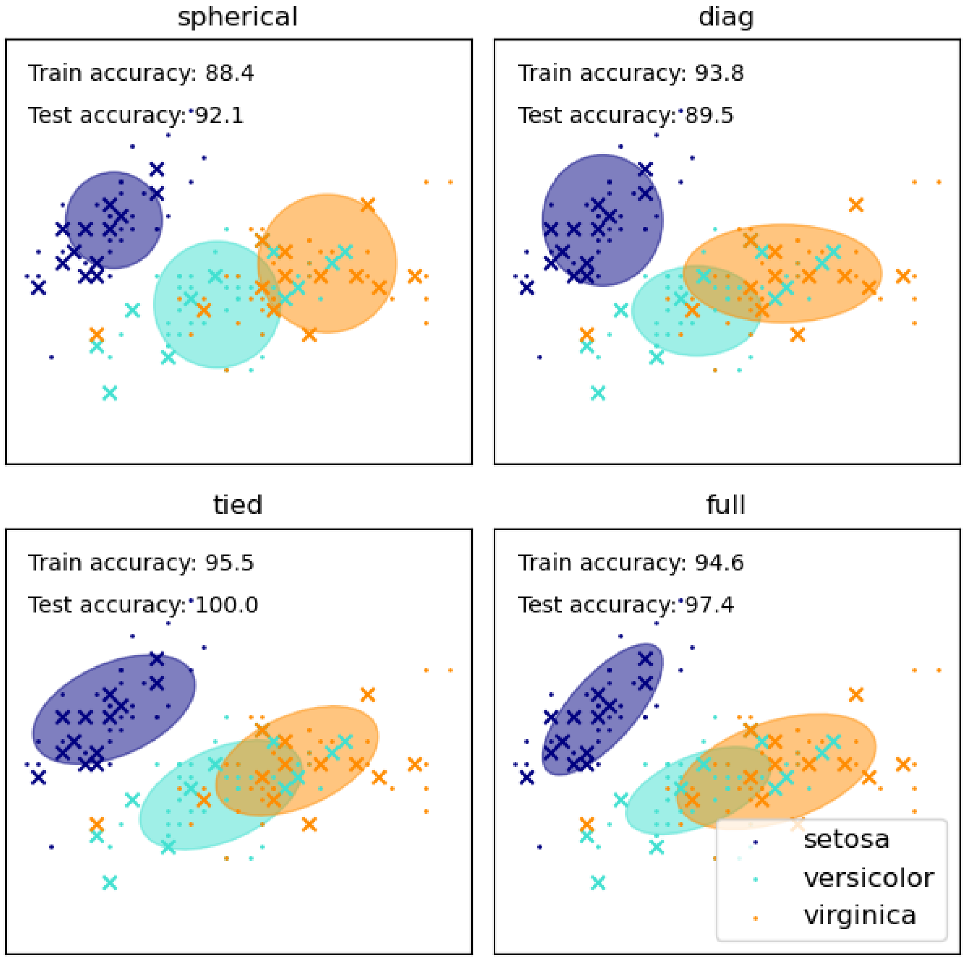
Full vs Diagonal



(a) Full covariance matrix ($t = 36$) (b) Diagonal covariance matrix ($t = 29$)

GMM in Sklearn

- [Additional options]() for covariance matrices include:
  - Spherical: Each cluster has a single variance (isotropic covariance)
  - Tied: All clusters share same covariance matrix



EM in d Dimensions

- Expectation step:

$$w_{ij} = P(C_i|\mathbf{x}_j) = \frac{f_i(\mathbf{x}_j) \cdot P(C_i)}{\sum_{a=1}^{k} f_a(\mathbf{x}_j) \cdot P(C_a)}$$

- Maximization step:

$$\boldsymbol{\mu}_i = \frac{\sum_{j=1}^{n} w_{ij} \cdot \mathbf{x}_j}{\sum_{j=1}^{n} w_{ij}} \qquad \Sigma_i = \frac{\sum_{j=1}^{n} w_{ij}(\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^{n} w_{ij}} \qquad P(C_i) = \frac{\sum_{j=1}^{n} w_{ij}}{n}$$

GMM EM Algorithm

TEXAS A&M UNIVERSITY Engineering

- Each step maximizes log-likelihood
- Iterate until convergence
  - Set maximum iterations or set threshold for changes in parameters
  - May converge to local optima

MLE:

$$\theta^* = \arg\max_{\theta}\{\ln P(\mathbf{D}|\theta)\}$$

Log-likelihood:

$$\ln P(\mathbf{D}|\theta) = \sum_{j=1}^{n} \ln f(\mathbf{x}_j) = \sum_{j=1}^{n} \ln\left(\sum_{i=1}^{k} f(\mathbf{x}_j|\boldsymbol{\mu}_i, \Sigma_i) P(C_i)\right)$$

GMM EM Algorithm Pseudocode

Expectation-Maximization (D, k, ϵ)**:**

$t \leftarrow 0$

Randomly initialize $\boldsymbol{\mu}_1^t, \ldots, \boldsymbol{\mu}_k^t$

$\Sigma_i^t \leftarrow \boldsymbol{I},\ \forall i = 1, \ldots, k$

repeat

    $t \leftarrow t + 1$

    for $i = 1, \ldots, k$ *and* $j = 1, \ldots, n$ **do**

        $w_{ij} \leftarrow \frac{f(\mathbf{x}_j | \boldsymbol{\mu}_i, \Sigma_i) \cdot P(C_i)}{\sum_{a=1}^{k} f(\mathbf{x}_j | \boldsymbol{\mu}_a, \Sigma_a) \cdot P(C_a)}$ `// posterior probability` $P^t(C_i | \mathbf{x}_j)$

    for $i = 1, \ldots, k$ **do**

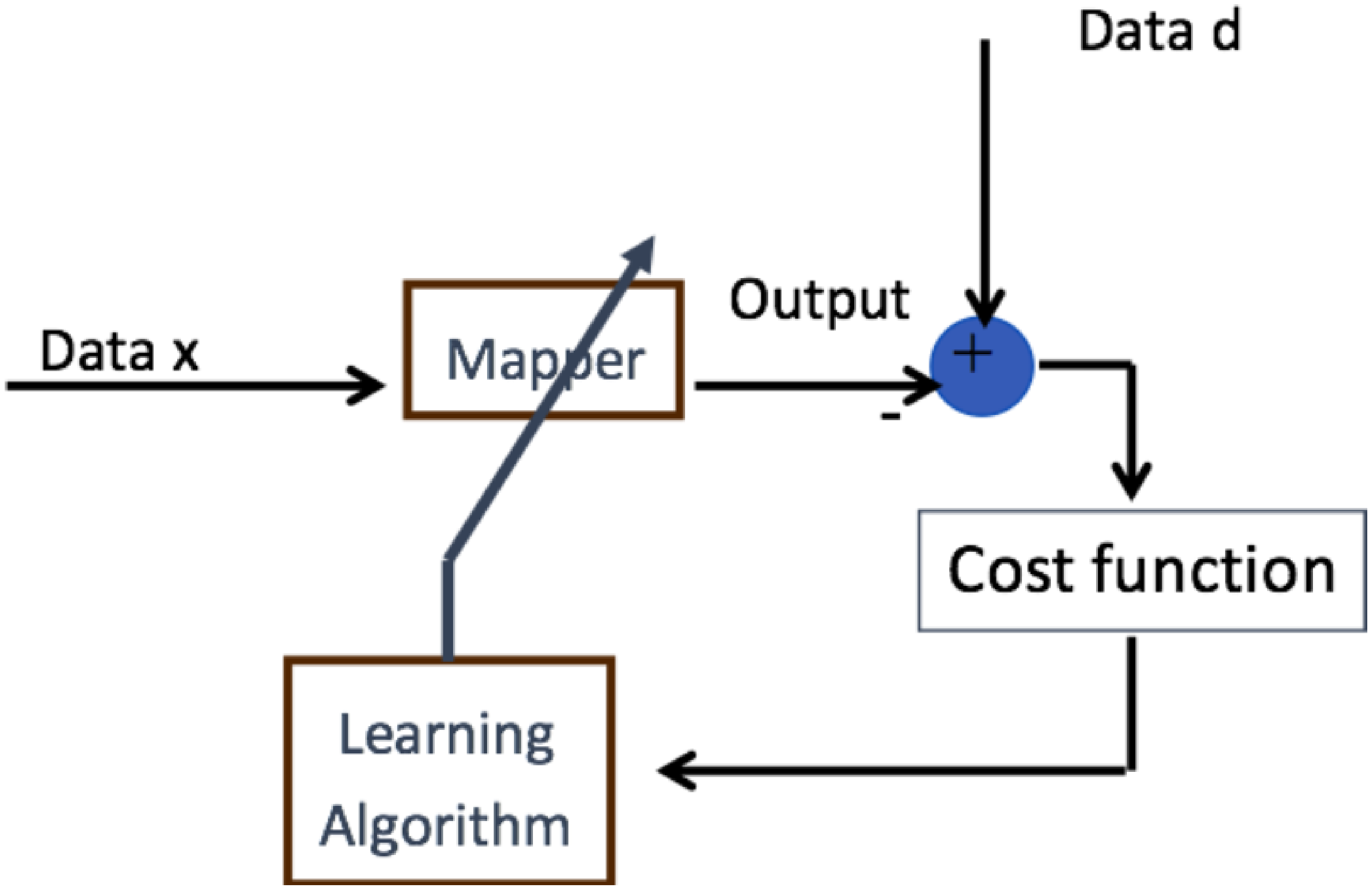
        $\boldsymbol{\mu}_i^t \leftarrow \frac{\sum_{j=1}^{n} w_{ij} \cdot \mathbf{x}_j}{\sum_{j=1}^{n} w_{ij}}$ `// re-estimate mean`

        $\Sigma_i^t \leftarrow \frac{\sum_{j=1}^{n} w_{ij} (\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^{n} w_{ij}}$ `// re-estimate covariance matrix`

        $P^t(C_i) \leftarrow \frac{\sum_{j=1}^{n} w_{ij}}{n}$ `// re-estimate priors`

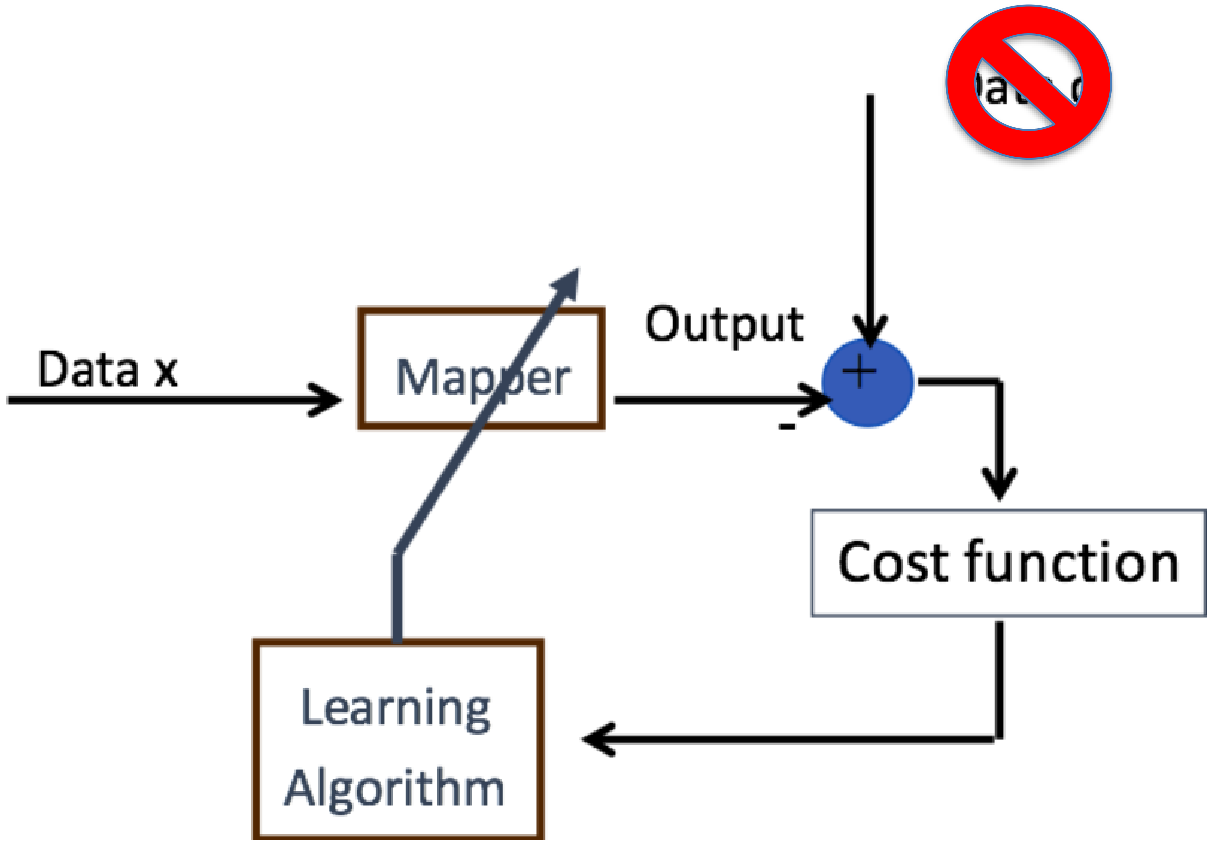
until $\sum_{i=1}^{k} \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|^2 \leq \epsilon$

GMM Machine Learning Model



GMM Machine Learning Model

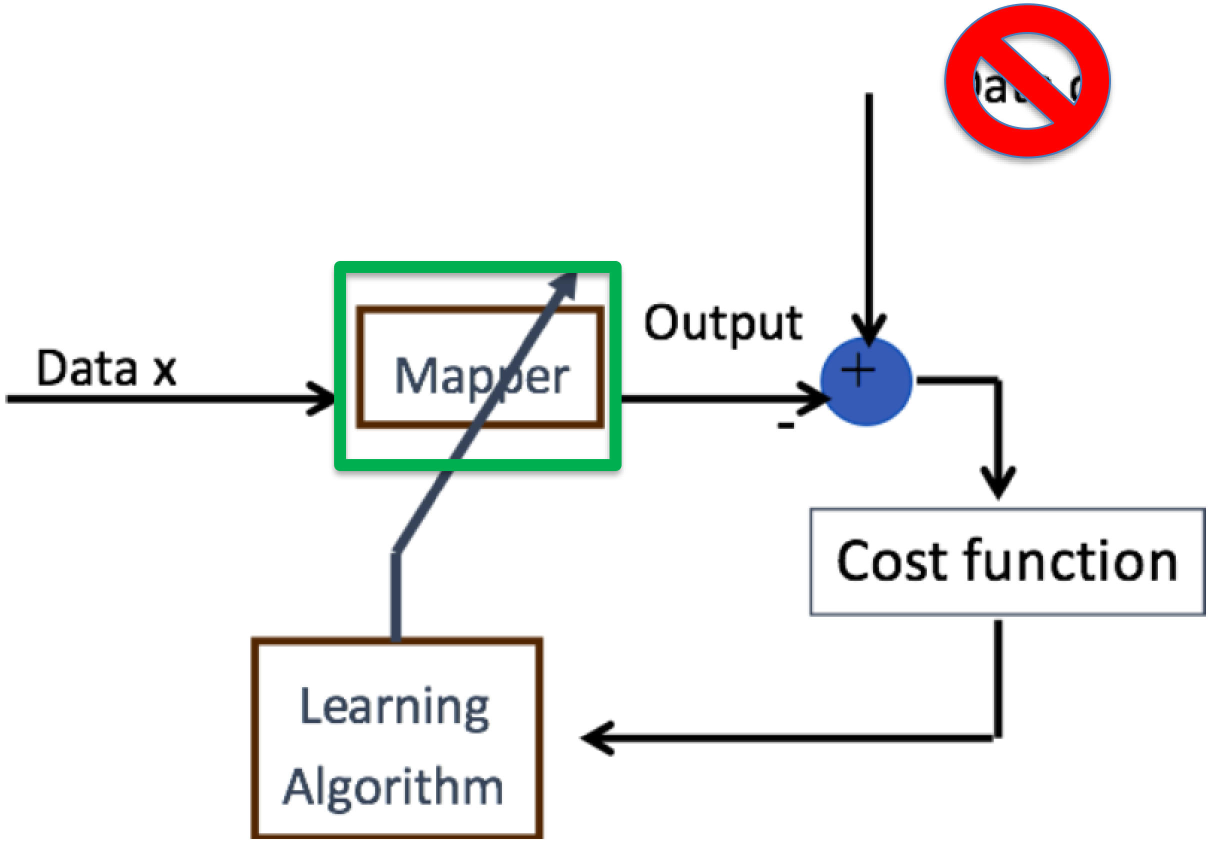
- Unsupervised: No labels, *d*



GMM Machine Learning Model

TEXAS A&M UNIVERSITY
Engineering

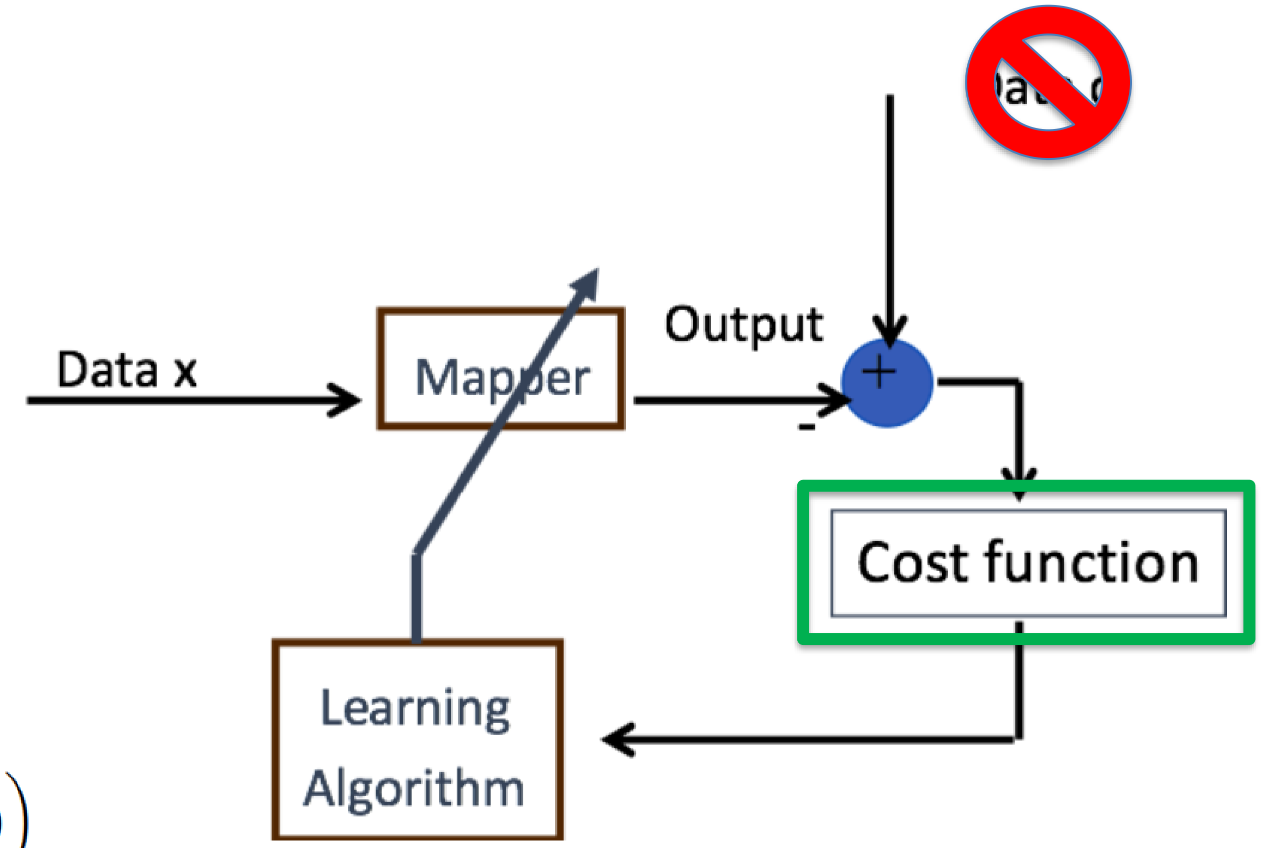
- Unsupervised: No labels, *d*
- **Mapper:**
  - GMM algorithm
  - Takes input data and groups into *k* clusters



GMM Machine Learning Model

- Unsupervised: No labels, *d*
- Mapper:
  - GMM algorithm
  - Takes input data and groups into *k* clusters
- **Cost function:**
  - Log-likelihood

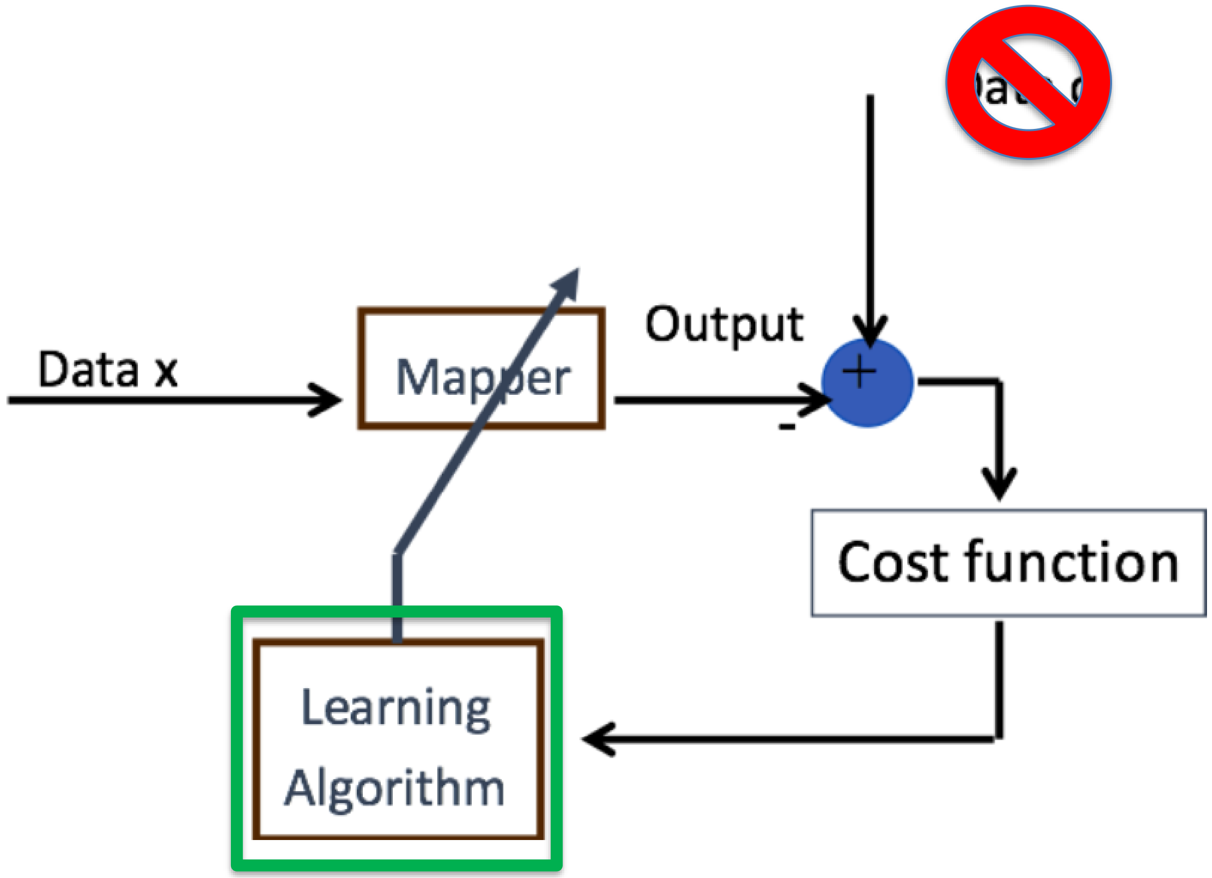
$$\ln P(\boldsymbol{D}|\theta) = \sum_{j=1}^{n} \ln f(\boldsymbol{x}_j) = \sum_{j=1}^{n} \ln\left(\sum_{i=1}^{k} f(\boldsymbol{x}_j|\boldsymbol{\mu}_i, \Sigma_i) P(C_i)\right)$$



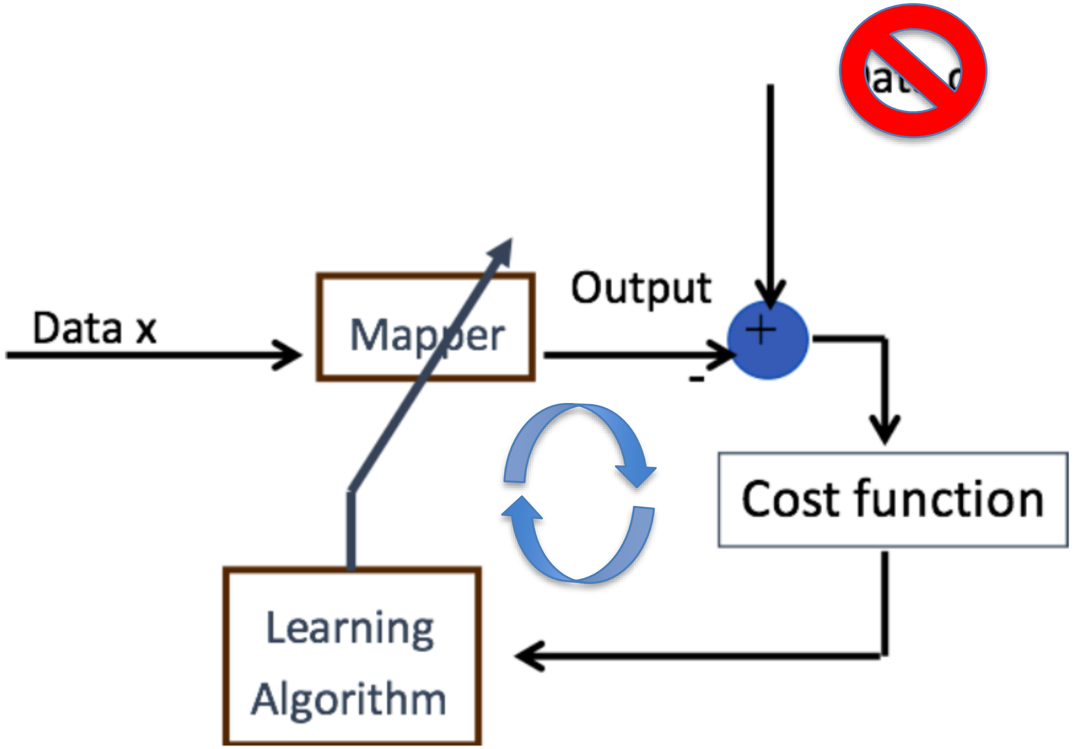
GMM Machine Learning Model

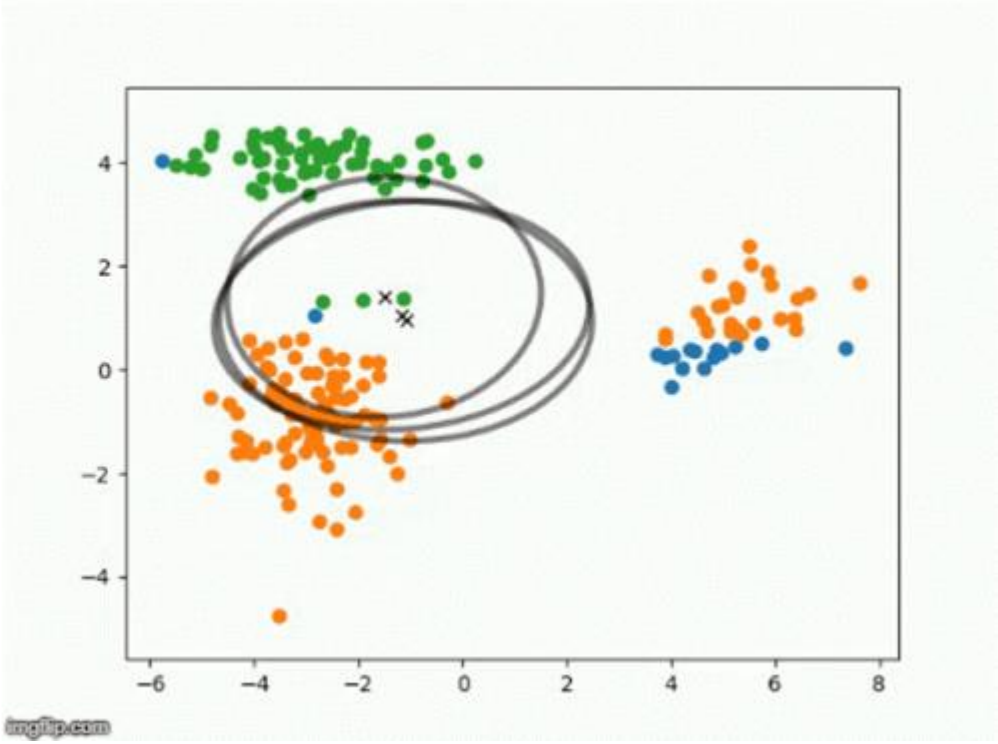
- Unsupervised: No labels, *d*
- Mapper:
  - GMM algorithm
  - Takes input data and groups into *k* clusters
- Cost function:
  - Sum of squared errors (SSE)
- **Learning algorithm**
  - MLE via EM approach

$$\theta^* = \arg\max_{\theta}\{\ln P(\boldsymbol{D}|\theta)\}$$



GMM Machine Learning Model





Gif from: Gaussian Mixture Models Em Method Math GIF

k-Means and EM Algorithm

k-Means and EM Algorithm

- Special case of EM algorithm
- What is the covariance matrix in the case of k-means?

$$P(\mathbf{x}_j|C_i) = \begin{cases} 1 & \text{if } C_i = \arg\min_{C_a} \left\{ \|\mathbf{x}_j - \boldsymbol{\mu}_a\|^2 \right\} \\ 0 & \text{otherwise} \end{cases}$$

$$P(C_i|\mathbf{x}_j) = \frac{P(\mathbf{x}_j|C_i)P(C_i)}{\sum_{a=1}^{k} P(\mathbf{x}_j|C_a)P(C_a)}$$

$$\boxed{P(C_i|\mathbf{x}_j) = \begin{cases} 1 & \text{if } \mathbf{x}_j \in C_i, \text{i.e., if } C_i = \arg\min_{C_a} \left\{ \|\mathbf{x}_j - \boldsymbol{\mu}_a\|^2 \right\} \\ 0 & \text{otherwise} \end{cases}}$$

How to choose number of clusters for GMM?

Choosing Number of Clusters/Components for GMMs

TEXAS A&M UNIVERSITY Engineering

- Two metrics:
  - Akaike Information Criterion (AIC)
  - Bayesian Information Criterion (BIC)
- Find balance between model complexity and goodness-of-fit
- Aim to minimize metrics

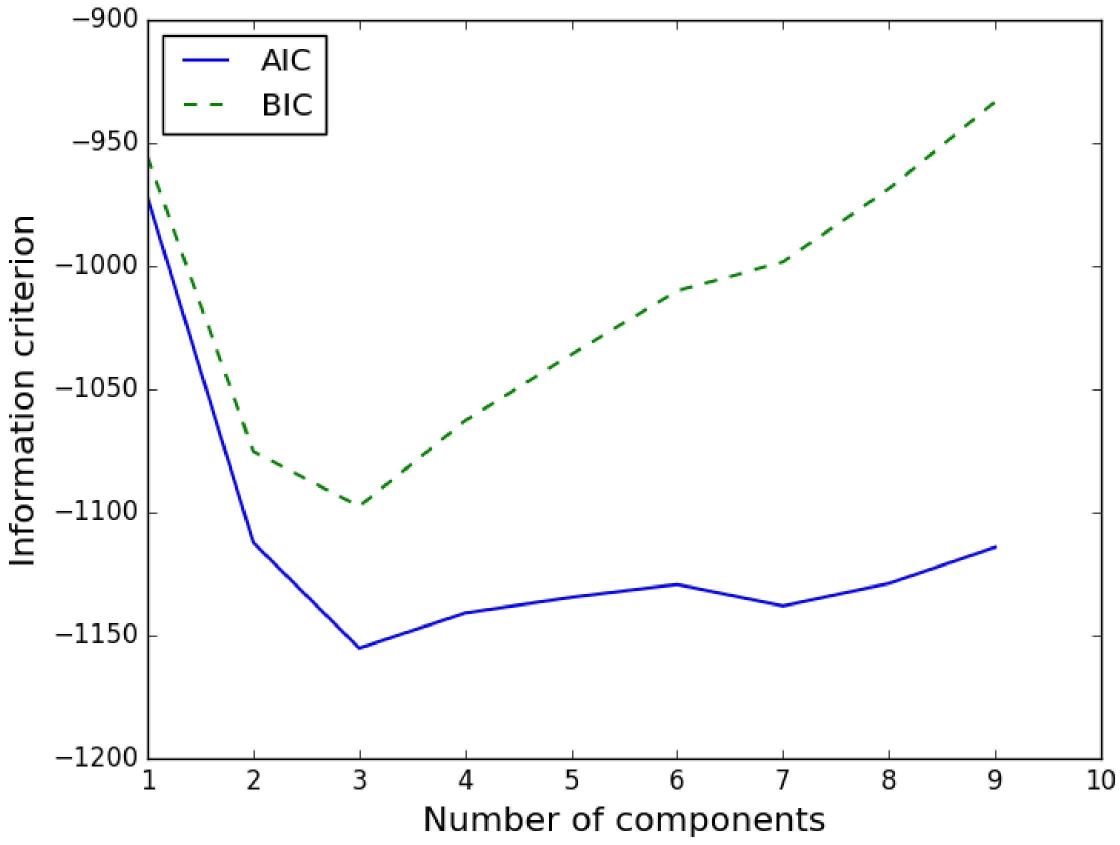


Image from: S. Kulkarni and S. Desai, Classifying Exoplanets with Gaussian Mixture Model

Choosing Number of Clusters/Components for GMMs

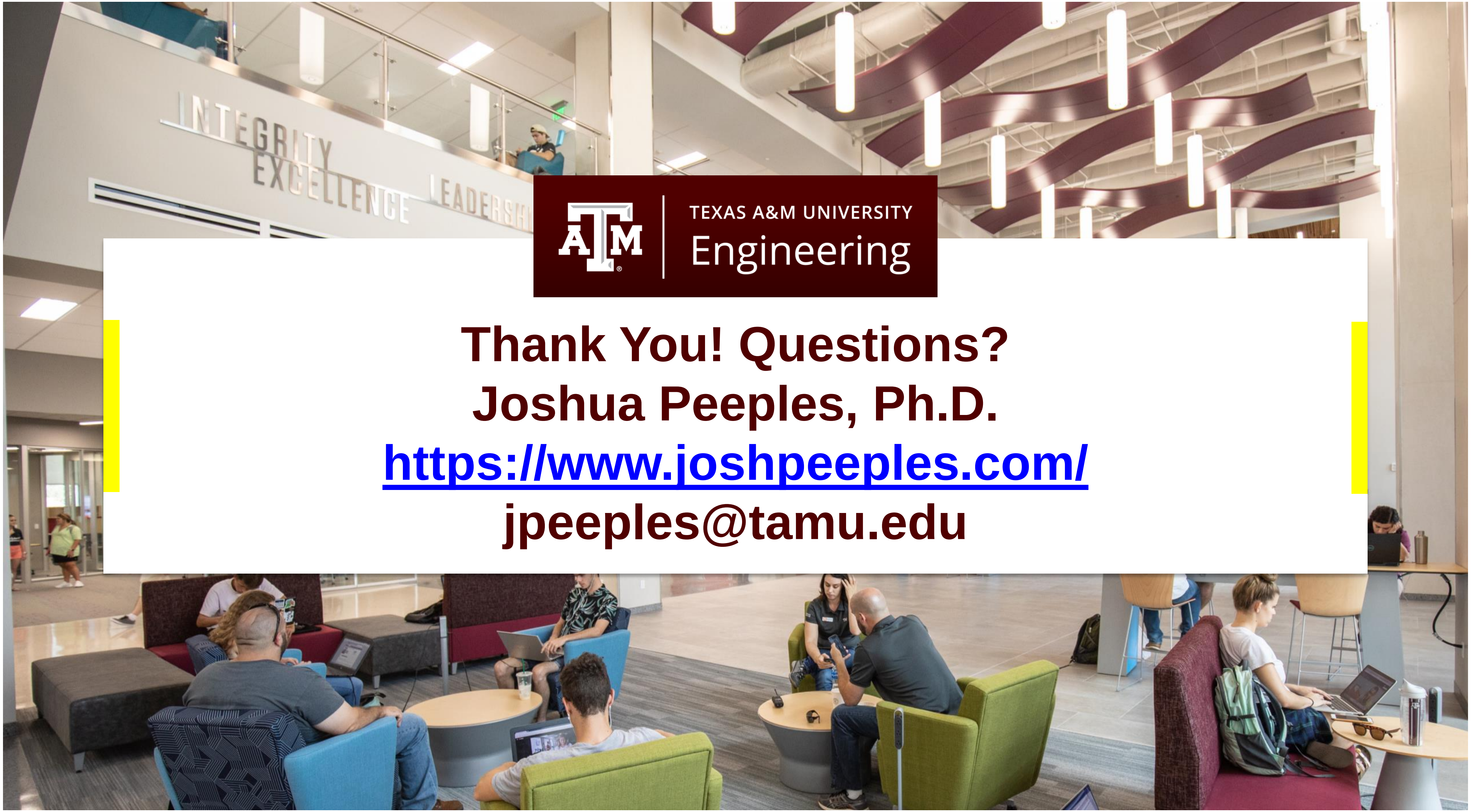
- Two metrics:
  - Akaike Information Criterion (AIC)
  - Bayesian Information Criterion (BIC)
- p is number of estimated parameters
- $\hat{L}$ is the maximized value of the likelihood function
- n is the number of samples

$$\mathrm{AIC} = 2p - 2\ln(\hat{L})$$

$$\mathrm{BIC} = p\ln(n) - 2\ln(\widehat{L})$$

Next class

- Hierarchical Clustering



TEXAS A&M UNIVERSITY
Engineering

Thank You! Questions?

Joshua Peeples, Ph.D.

https://www.joshpeeples.com/

jpeeples@tamu.edu

TEXAS A&M UNIVERSITY
Engineering

Supplemental Slides

Useful Links

TEXAS A&M UNIVERSITY Engineering

- Gaussian Mixture Models and EM
- Gaussian Mixture Models Google Colab
- Bayes Theorem Clearly Explained
- Maximum Likelihood Clearly Explained
- Maximum Likelihood Estimation of a Coin Flip
- Parameter Estimation (MLE)