

# Unsupervised Learning

CSCC11 – Topic 06



Computer & Mathematical Sciences  
**UNIVERSITY OF TORONTO**  
SCARBOROUGH

# Topics

- In Unsupervised Learning, the dataset contains unlabeled data
- The goal is to model the intrinsic structure of the data
- Dimension Reduction
  - PCA
  - Probability PCA
- Clustering
  - K Means
  - K Means ++
  - Gaussian Mixture Models
  - EM Algorithm
- Density Estimation
  - PPCA
  - GMM

# Dimensionality Reduction

- Data very often are high dimensional
  - Each pixel in an image (greyscale) is a dimension.
  - An image can have millions of pixels.
  - Nearby pixels are very similar, hence **highly correlated**, or **redundant**.
- How can we represent this image in the low dimension
  - **Removing redundancy** in the data
- Visualization of high dimensional data
- Preprocessing data
  - Before we perform regression/classification, first reduce the dimension of the data, then it will make the model have less parameters to learn.
- Compression: less storage, smaller amount of data to transmit
- Prior Modeling: learn a low dimensional Gaussian model for data

# Clustering

- Look for natural grouping of similar data
- Examples:
  - Image Segmentation
    - Partitioning an image into distinct regions for object recognition.
  - Document Classification
    - Grouping similar documents based on contents to retrieve information effectively.
  - Customer Segmentation
    - Grouping customers with similar purchasing behaviors to tailor marketing efforts.
  - Social Network Analysis
    - Identifying communities or groups of similar individuals within a network.

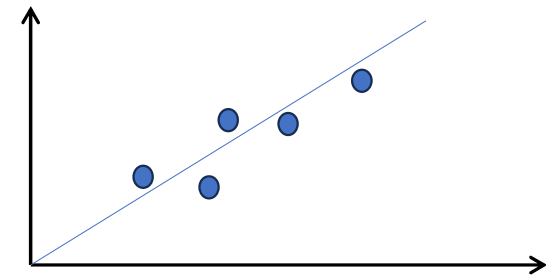
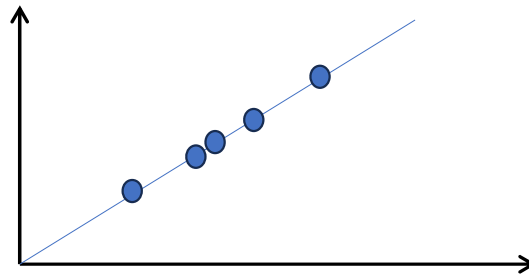
# Density Estimation

- Estimating the probability distribution that likely generated a given dataset
- Identify regions where data points are dense (high probability) and regions where data points are sparse (low probability)
- Example: given a large set of images, find the probability distribution such that it gives high probability to images that look like the images in the training set and low probability to those images that do not like the images in the training set.
- Applications
  - Image processing/restoration, image in-painting, super resolution task
  - Anomaly detection

# Principal Component Analysis

# Dimensionality Reduction

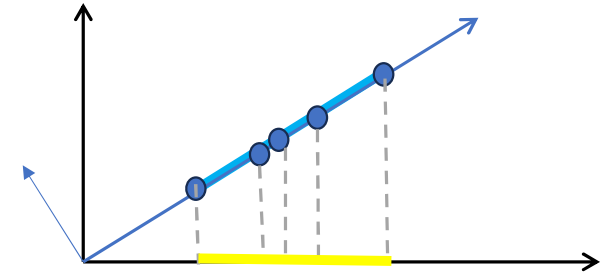
- Very often, data are high-dimensional (e.g. images)
  - This is our observation space
- Goal: find a low-dimensional representation of high-dimensional data
  - The low-dimensional representation is also called **latent embedding**
  - Find the **mapping** from the **observation space** to the **latent embedding space** while **preserving** as much **information** as possible
- Picture



# Variation of Data

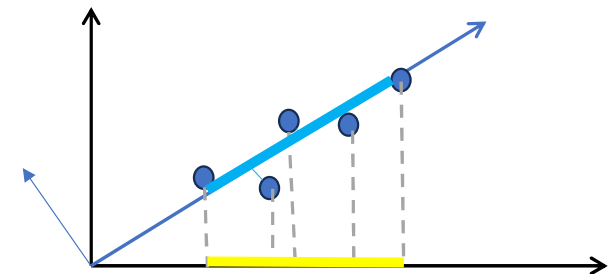
- Example 1

- We are rotating our coordinate system
- In this new coordinate system, we can use a line to represent the data
- The new coordinate system maximizes the variation of the data



- Example 2

- Data are close to a line but not exactly on a line
- Project data to the rotated coordinate system
- The line minimizes the perpendicular distances to the points maximizes the variation of data





# Projecting a Vector onto another Vector

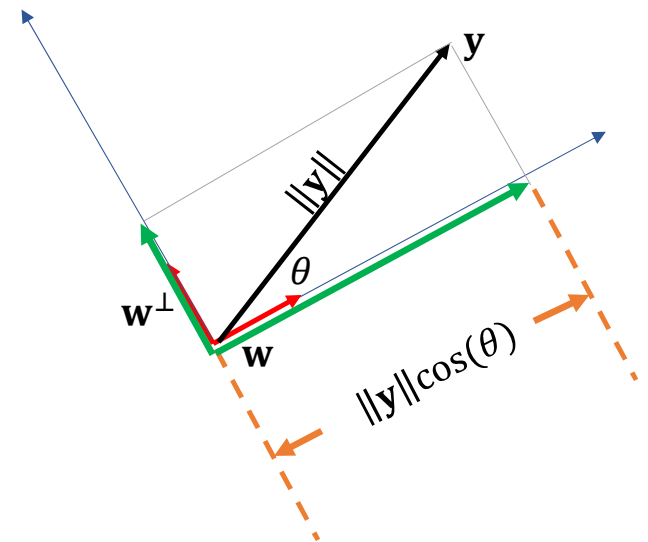
- Let  $\mathbf{w}, \mathbf{w}^\perp \in \mathbb{R}^2$ , where  $\|\mathbf{w}\| = \|\mathbf{w}^\perp\| = 1$
- Let  $\mathbf{y} \in \mathbb{R}^2$  be a vector
- The component of  $\mathbf{y}$  in the direction of  $\mathbf{w}$  is

$$\mathbf{y}_\mathbf{w} = (\mathbf{y}^T \mathbf{w}) \mathbf{w}$$

- The component of  $\mathbf{y}$  in the direction of  $\mathbf{w}^\perp$  is

$$\mathbf{y}_{\mathbf{w}^\perp} = (\mathbf{y}^T \mathbf{w}^\perp) \mathbf{w}^\perp$$

$$\begin{aligned} \mathbf{y}^T \mathbf{w} &= \|\mathbf{y}\| \|\mathbf{w}\| \cos(\theta) \\ &= \|\mathbf{y}\| \cos(\theta) \end{aligned}$$



# Principle Component Analysis

- Goal of PCA is to find a **linear mapping** from the high-dimensional space to the low-dimensional space while **preserving** maximum possible amount of **information**.
- PCA is to **project** our high dimensional data onto a linear subspace that maximizes the variance of projected data

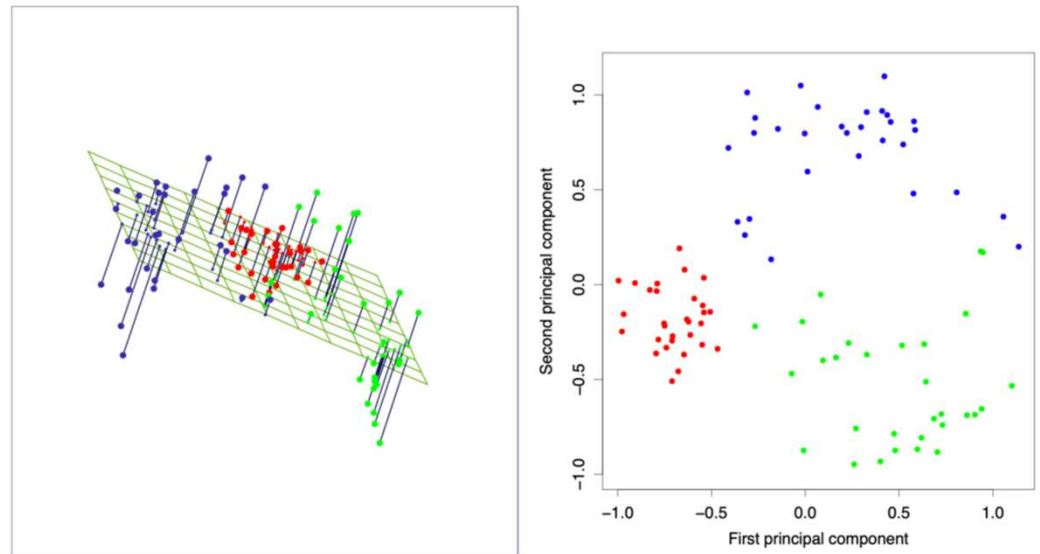
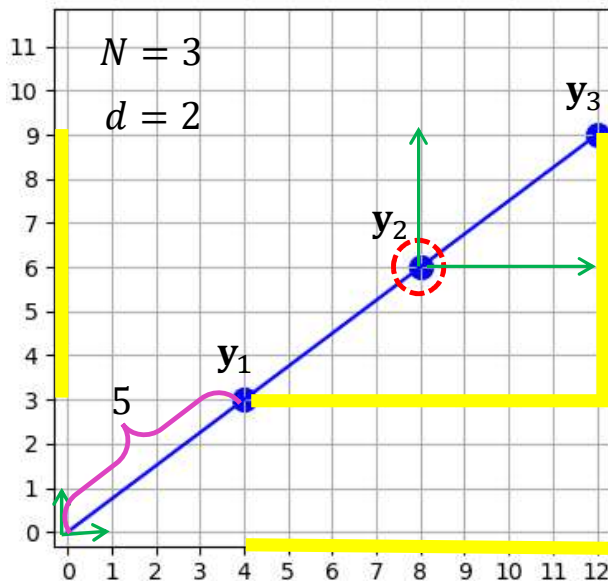


Image Courtesy of [HTF]

# Variance of 2-D Data



$$\mathbf{y}_1 = \begin{bmatrix} y_{11} \\ y_{21} \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

$$\mathbf{y}_2 = \begin{bmatrix} y_{12} \\ y_{22} \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

$$\mathbf{y}_3 = \begin{bmatrix} y_{13} \\ y_{23} \end{bmatrix} = \begin{bmatrix} 12 \\ 9 \end{bmatrix}$$

$$\boldsymbol{\mu}_Y = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_i = \frac{1}{3} \begin{bmatrix} 4 + 8 + 12 \\ 3 + 6 + 9 \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} | & | & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \\ | & | & | \end{bmatrix}_{2 \times 3}$$

$$\mathbf{Y} = \begin{bmatrix} 4 & 8 & 12 \\ 3 & 6 & 9 \end{bmatrix}_{2 \times 3}$$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [\mathbf{e}_1 \quad \mathbf{e}_2]$$

$$\mathbf{1} = \mathbf{1}_3 = [1 \quad 1 \quad 1]_{1 \times 3}$$

$$\mathbf{Z} = \mathbf{Y} - \boldsymbol{\mu}_Y \mathbf{1}$$

$$\mathbf{Z} = \begin{bmatrix} -4 & 0 & 4 \\ -3 & 0 & 3 \end{bmatrix}$$

$$\boldsymbol{\mu}_Z = \frac{1}{N} \sum_{i=1}^N \mathbf{z}_i = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{S} \equiv \mathbf{S}_Y = \frac{1}{N-1} (\mathbf{Y} - \boldsymbol{\mu}_Y \mathbf{1})(\mathbf{Y} - \boldsymbol{\mu}_Y \mathbf{1})^T = \frac{1}{N-1} \mathbf{Z} \mathbf{Z}^T = \mathbf{S}_Z = \frac{1}{2} \begin{bmatrix} -4 & 0 & 4 \\ -3 & 0 & 3 \end{bmatrix} \begin{bmatrix} -4 & -3 \\ 0 & 0 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 16 & 12 \\ 12 & 9 \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_1 \sigma_2 \\ \sigma_2 \sigma_1 & \sigma_2^2 \end{bmatrix}$$

$\mathbf{S}$  is the sample covariance, hence divided by  $(N - 1)$

You can also use population covariance, which is divided by  $N$  for deriving PCA

$$\mathbf{e}_i^T \mathbf{S} \mathbf{e}_i = \sigma_i^2$$

# Remarks

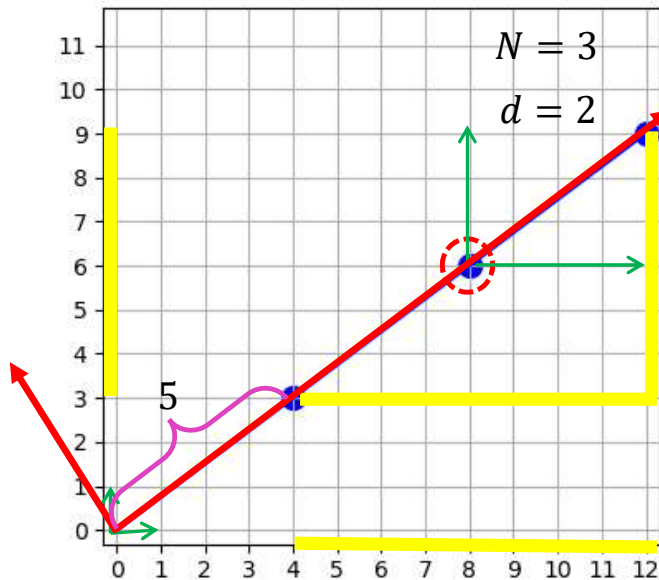
- Centering the data only shifts the coordinate system without rotation. It does not change the covariance matrix of the original data
- The variance of the data in one direction in the coordinate system is the variance of all the data projected onto that direction.

$$\text{Var}(\mathbf{e}_i^T \mathbf{Y}) = \frac{1}{N-1} \|\mathbf{e}_i^T \mathbf{Y} - \mathbf{e}_i^T \boldsymbol{\mu}_Y \mathbf{1}\|_2^2 = \frac{1}{N-1} (\mathbf{e}_i^T (\mathbf{Y} - \boldsymbol{\mu}_Y \mathbf{1}) (\mathbf{Y} - \boldsymbol{\mu}_Y \mathbf{1})^T \mathbf{e}_i) = \mathbf{e}_i^T \mathbf{S} \mathbf{e}_i = \sigma_i^2$$

- The diagonals of the covariance matrix contain all the variances of data in each direction of the coordinate system
- The sum of the diagonals (i.e.  $\text{Tr}(\mathbf{S})$ ) is the **variance of the dataset**

# Rotate the Coordinate System

$$\mathbf{v}_i^T \mathbf{v}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad \mathbf{V} = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} \\ \frac{3}{5} & \frac{4}{5} \end{bmatrix} = [\mathbf{v}_1 \quad \mathbf{v}_2]$$



$$\mathbf{y}_1 = \begin{bmatrix} y_{11} \\ y_{21} \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} | & | & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \\ | & | & | \end{bmatrix}_{2 \times 3}$$

$$\mathbf{y}_2 = \begin{bmatrix} y_{12} \\ y_{22} \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 4 & 8 & 12 \\ 3 & 6 & 9 \end{bmatrix}_{2 \times 3}$$

$$\mathbf{y}_3 = \begin{bmatrix} y_{13} \\ y_{23} \end{bmatrix} = \begin{bmatrix} 12 \\ 9 \end{bmatrix}$$

$$\boldsymbol{\mu}_Y = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_i = \frac{1}{3} \begin{bmatrix} 4 + 8 + 12 \\ 3 + 6 + 9 \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

$$\mathbf{1} = \mathbf{1}_3 = [1 \quad 1 \quad 1]_{1 \times 3}$$

$$\mathbf{A} = \mathbf{V}^T \mathbf{Y} = \begin{bmatrix} 5 & 10 & 15 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{\mu}_A = \frac{1}{N} \sum_{i=1}^N \mathbf{a}_i = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

$$\mathbf{X} = \mathbf{V}^T \mathbf{Y} - \boldsymbol{\mu}_A \mathbf{1}$$

$$\mathbf{X} = \begin{bmatrix} -5 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{S}_A = \frac{1}{N-1} (\mathbf{V}^T \mathbf{Y} - \boldsymbol{\mu}_A \mathbf{1})(\mathbf{V}^T \mathbf{Y} - \boldsymbol{\mu}_A \mathbf{1})^T = \frac{1}{N-1} \mathbf{X} \mathbf{X}^T = \mathbf{S}_X = \frac{1}{2} \begin{bmatrix} -5 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -5 & 0 \\ 0 & 0 \\ 5 & 0 \end{bmatrix} = \begin{bmatrix} 25 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_1 \sigma_2 \\ \sigma_2 \sigma_1 & \sigma_2^2 \end{bmatrix}$$

$\mathbf{V}$  is an orthogonal matrix, it has orthonormal basis  $\rightarrow \mathbf{V}^T \mathbf{V} = \mathbf{I}_d$

## Remark about $\mathbf{A}$

- What is the relationship between  $\mu_A$  and  $\mu_Y$  ?

$$\mu_A = E[\mathbf{A}] = E(\mathbf{V}^T \mathbf{Y}) = \mathbf{V}^T E(\mathbf{Y}) = \mathbf{V}^T \mu_Y$$

- What is the relationship between  $\mathbf{S}_A$  and  $\mathbf{S}_Y$

$$\begin{aligned}\mathbf{S}_A &= \frac{1}{N-1} (\mathbf{V}^T \mathbf{Y} - \mu_A \mathbf{1})(\mathbf{V}^T \mathbf{Y} - \mu_A \mathbf{1})^T \\ &= \frac{1}{N-1} (\mathbf{V}^T \mathbf{Y} - \mathbf{V}^T \mu_Y \mathbf{1})(\mathbf{V}^T \mathbf{Y} - \mathbf{V}^T \mu_Y \mathbf{1})^T \\ &= \mathbf{V}^T \frac{1}{N-1} (\mathbf{Y} - \mu_Y \mathbf{1})(\mathbf{Y} - \mu_Y \mathbf{1})^T \mathbf{V} \\ &= \mathbf{V}^T \mathbf{S}_Y \mathbf{V}\end{aligned}$$

$$\mathbf{A} = \mathbf{V}^T \mathbf{Y} = \begin{bmatrix} 5 & 10 & 15 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{X} = \mathbf{V}^T \mathbf{Y} - \mu_A \mathbf{1}$$

$$\mathbf{v}_i^T \mathbf{v}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$$\mu_A = \mathbf{V}^T \mu_Y$$

$$\mathbf{S}_A = \mathbf{V}^T \mathbf{S}_Y \mathbf{V}$$

# Trace of the Matrix after Rotation

- Rotating a coordinate system is to rotate a matrix in the original coordinate system (in the opposite direction).
- Rotating a matrix does not change the trace of the matrix

$$\text{Tr}(\mathbf{S}_A) = \text{Tr}(\mathbf{V}^T \mathbf{S}_Y \mathbf{V}) = \text{Tr}(\mathbf{S}_Y \mathbf{V} \mathbf{V}^T) = \text{Tr}(\mathbf{S}_Y \mathbf{I}_d) = \text{Tr}(\mathbf{S}_Y)$$



$$\text{Tr}(\mathbf{S}_A) = \text{Tr}(\mathbf{S}_Y)$$

Cyclic Property of the Trace

- In the rotated coordinate system, the variance dataset does not change

## Remark about $\mathbf{X}$ and $\mathbf{Z}$

- $\mathbf{Z}$  is centered data under the original coordinate system
- $\mathbf{X}$  is centered data under the rotated coordinate system

$$\mathbf{X} = \mathbf{V}^T \mathbf{Y} - E[\mathbf{V}^T \mathbf{Y}] \mathbf{1} = \mathbf{V}^T (\mathbf{Y} - E[\mathbf{V}^T \mathbf{Y}] \mathbf{1}) = \mathbf{V}^T (\mathbf{Y} - \mu_Y \mathbf{1}) = \mathbf{V}^T \mathbf{Z}$$



$$\mathbf{X} = \mathbf{V}^T \mathbf{Z}$$

$$\mu_X = E[\mathbf{X}] = E(\mathbf{V}^T \mathbf{Y} - \mathbf{V}^T \mu_Y \mathbf{1}) = \mathbf{V}^T E(\mathbf{Y} - \mu_Y \mathbf{1}) = \mathbf{V}^T (\mu_Y - \mu_Y) = \mathbf{0}$$

$$\mu_X = \mathbf{V}^T \mu_Z = \mathbf{0}$$

$$\mathbf{S}_X = \mathbf{V}^T \mathbf{S}_Z \mathbf{V} = \mathbf{V}^T \mathbf{S}_Y \mathbf{V}$$



# Dimension Reduction

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [\mathbf{e}_1 \quad \mathbf{e}_2]$$

$$\mathbf{Y} = \begin{bmatrix} 4 & 8 & 12 \\ 3 & 6 & 9 \end{bmatrix}$$

$$\boldsymbol{\mu}_Y = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

$$\mathbf{Z} = \mathbf{Y} - \boldsymbol{\mu}_Y \mathbf{1}$$

$$= \begin{bmatrix} -4 & 0 & 4 \\ -3 & 0 & 3 \end{bmatrix}$$

$$\boldsymbol{\mu}_Z = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{S} \equiv \mathbf{S}_Y = \mathbf{S}_Z = \begin{bmatrix} 16 & 12 \\ 12 & 9 \end{bmatrix}$$

$$\mathbf{V} = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} \\ \frac{3}{5} & \frac{4}{5} \end{bmatrix} = [\mathbf{v}_1 \quad \mathbf{v}_2]$$

$$\mathbf{A} = \mathbf{V}^T \mathbf{Y} = \begin{bmatrix} 5 & 10 & 15 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{\mu}_A = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$

$$\mathbf{X} = \mathbf{V}^T \mathbf{Y} - \boldsymbol{\mu}_A = \mathbf{V}^T \mathbf{Z}$$

$$= \begin{bmatrix} -5 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

$$\boldsymbol{\mu}_X = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{S}_A = \mathbf{S}_X = \begin{bmatrix} 25 & 0 \\ 0 & 0 \end{bmatrix}$$

- The variance in the direction of  $\mathbf{v}_2$  is zero
- There is no information in the direction of  $\mathbf{v}_2$

$$\mathbf{W} = \begin{bmatrix} \frac{4}{5} \\ \frac{3}{5} \end{bmatrix} = [\mathbf{v}_1]$$

$$\hat{\mathbf{X}} = [\mathbf{x}_1] = \mathbf{W}^T (\mathbf{Y} - \boldsymbol{\mu}_Y)$$

$$= \begin{bmatrix} \frac{4}{5} & \frac{3}{5} \end{bmatrix} \begin{bmatrix} -4 & 0 & 4 \\ -3 & 0 & 3 \end{bmatrix}$$

$$= [-5 \quad 0 \quad 5]$$

$\hat{\mathbf{X}}$  is  $k \times N$ , where  $1 = k < d = 2$   
 $\mathbf{W}$  is  $d \times k$

# Reconstruct $\mathbf{Y}$

$$\mathbf{X} = \mathbf{V}^T(\mathbf{Y} - \mu_Y \mathbf{1})$$

$$\mathbf{V}\mathbf{X} = \mathbf{V}\mathbf{V}^T(\mathbf{Y} - \mu_Y \mathbf{1})$$

$$\mathbf{V}\mathbf{V}^T = \mathbf{I}$$

$$\mathbf{V}\mathbf{X} = \mathbf{Y} - \mu_Y \mathbf{1}$$

$$\text{Let } \mathbf{V} = [\mathbf{W} : \mathbf{W}^\perp] \quad \text{and } \mathbf{X} = \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{X}}^\perp \end{bmatrix}$$

$$\mathbf{Y} - \mu_Y \mathbf{1} = [\mathbf{W} | \mathbf{W}^\perp] \begin{bmatrix} \hat{\mathbf{X}} \\ - \end{bmatrix}$$

$$\mathbf{Y} = \mathbf{W}\hat{\mathbf{X}} + \mathbf{W}^\perp \hat{\mathbf{X}}^\perp + \mu_Y \mathbf{1}$$

$$= \mathbf{W}\hat{\mathbf{X}} + \mu_Y \mathbf{1} + \mathbf{W}^\perp \hat{\mathbf{X}}^\perp$$

$$\uparrow$$

$$\hat{\mathbf{Y}}$$

$$\mathbf{Y} = \begin{bmatrix} 4 & 8 & 12 \\ 3 & 6 & 9 \end{bmatrix} \quad \mu_Y = \begin{bmatrix} 8 \\ 6 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} -5 & 0 & 5 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{X}}^\perp \end{bmatrix}$$

$$\mathbf{Y} \in \mathbb{R}^{d \times N}$$

$$\mu_Y \in \mathbb{R}^{d \times 1}$$

$$\mathbf{W} \in \mathbb{R}^{d \times k}$$

$$\hat{\mathbf{X}} \in \mathbb{R}^{k \times N}$$

$$\mathbf{W}^\perp \in \mathbb{R}^{d \times (d-k)}$$

$$\hat{\mathbf{X}}^\perp \in \mathbb{R}^{(d-k) \times N}$$

$$\mathbf{V} = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} \\ \frac{3}{5} & \frac{4}{5} \\ \frac{3}{5} & \frac{4}{5} \end{bmatrix} = [\mathbf{v}_1 \quad \mathbf{v}_2] = [\mathbf{W} | \mathbf{W}^\perp]$$

$$\mathbf{W} = \begin{bmatrix} \frac{4}{5} \\ \frac{3}{5} \\ \frac{3}{5} \end{bmatrix} = [\mathbf{v}_1]$$

$$\mathbf{W}^\perp = \begin{bmatrix} -\frac{3}{5} \\ \frac{4}{5} \\ \frac{4}{5} \end{bmatrix} = [\mathbf{v}_2]$$

$$\hat{\mathbf{X}} = \begin{bmatrix} -5 & 0 & 5 \end{bmatrix}$$

$$\hat{\mathbf{X}}^\perp = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{W}\hat{\mathbf{X}} = \begin{bmatrix} -4 & 0 & 4 \\ -3 & 0 & 3 \end{bmatrix}$$

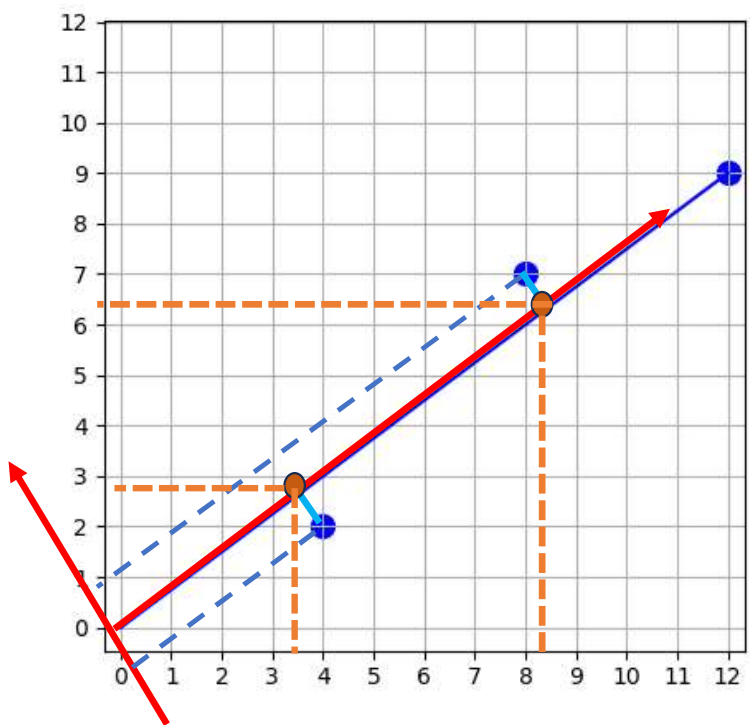
$$\mathbf{W}^\perp \hat{\mathbf{X}}^\perp = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\hat{\mathbf{Y}} = \mathbf{W}\hat{\mathbf{X}} + \mu_Y \mathbf{1} = \begin{bmatrix} 4 & 8 & 12 \\ 3 & 6 & 9 \end{bmatrix} = \mathbf{Y}$$

$$\mathbf{V} = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} \\ \frac{3}{5} & \frac{4}{5} \end{bmatrix} = [\mathbf{v}_1 \quad \mathbf{v}_2] \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

$$\hat{\mathbf{X}} = [\mathbf{x}_1] \\ \hat{\mathbf{X}}^\perp = [\mathbf{x}_2]$$

$$\mathbf{W} = \begin{bmatrix} \frac{4}{5} \\ \frac{3}{5} \end{bmatrix} = [\mathbf{v}_1] \quad \mathbf{W}^\perp = \begin{bmatrix} -\frac{3}{5} \\ \frac{4}{5} \end{bmatrix} = [\mathbf{v}_2]$$



$$\mathbf{Y} = \begin{bmatrix} 4 & 8 & 12 \\ 2 & 7 & 9 \end{bmatrix}$$

$$\mu_Y = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix} -4 & 0 & 4 \\ -4 & 1 & 3 \end{bmatrix}$$

$$\mathbf{S} \equiv \mathbf{S}_Y = \begin{bmatrix} 16 & 14 \\ 14 & 13 \end{bmatrix}$$

$$\mathbf{S}_X = \begin{bmatrix} 28.36 & 2.48 \\ 2.48 & 0.64 \end{bmatrix}$$

$$\mathbf{X} = \mathbf{V}^T (\mathbf{Y} - \mu_Y \mathbf{1}) = \mathbf{V}^T \mathbf{Z}$$

$$= \begin{bmatrix} -5.6 & 0.6 & 5 \\ -0.8 & 0.8 & 0 \end{bmatrix}$$

$$\hat{\mathbf{X}} = \begin{bmatrix} -5.6 & 0.6 & 5 \end{bmatrix}$$

$$\mathbf{W}\hat{\mathbf{X}} = \begin{bmatrix} -4.48 & 0.48 & 4 \\ -3.36 & 0.36 & 3 \end{bmatrix}$$

$$\hat{\mathbf{X}}^\perp = \begin{bmatrix} -0.8 & 0.8 & 0 \end{bmatrix}$$

$$\mathbf{W}^\perp \hat{\mathbf{X}}^\perp = \begin{bmatrix} 0.48 & -0.48 & 0 \\ -0.64 & 0.64 & 0 \end{bmatrix}$$

$$\hat{\mathbf{Y}} = \mathbf{W}\hat{\mathbf{X}} + \mu_Y \mathbf{1} = \begin{bmatrix} 3.52 & 8.48 & 12 \\ 2.64 & 6.36 & 9 \end{bmatrix} = \mathbf{Y} - (\mathbf{W}^\perp \hat{\mathbf{X}}^\perp) = \begin{bmatrix} 4 & 8 & 12 \\ 2 & 7 & 9 \end{bmatrix} - \begin{bmatrix} 0.48 & -0.48 & 0 \\ -0.64 & 0.64 & 0 \end{bmatrix}$$

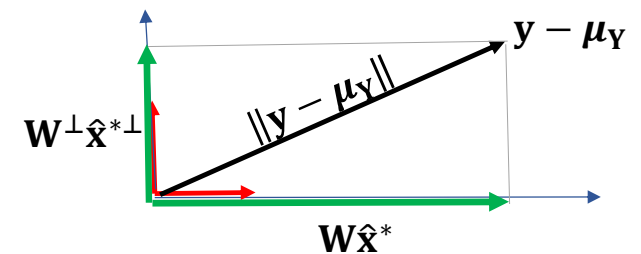
# Remarks

- $\mathbf{W}^\perp \hat{\mathbf{X}}^\perp$  is the reconstruction error matrix

$$\hat{\mathbf{Y}} = \mathbf{W}\hat{\mathbf{X}} + \mu_{\mathbf{Y}}\mathbf{1} = \mathbf{Y} - \mathbf{W}^\perp \hat{\mathbf{X}}^\perp \quad \Rightarrow \quad \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{W}^\perp \hat{\mathbf{X}}^\perp$$

- Given a new point  $\hat{\mathbf{x}}^* \in \mathbb{R}^{k \times 1}$ , project back to reconstruct  $\hat{\mathbf{y}}^* \in \mathbb{R}^{d \times 1}$

$$\hat{\mathbf{y}}^* = \mathbf{W}\hat{\mathbf{x}}^* + \mu_{\mathbf{Y}} \Rightarrow \begin{cases} \mathbf{y} - \hat{\mathbf{y}}^* = \mathbf{W}^\perp \hat{\mathbf{x}}^{*\perp} \\ \hat{\mathbf{y}}^* - \mu_{\mathbf{Y}} = \mathbf{W}\hat{\mathbf{x}}^* \end{cases}$$



- Pythagorean Theorem:

$$(\mathbf{W}\hat{\mathbf{x}}^*) \perp (\mathbf{W}^\perp \hat{\mathbf{x}}^{*\perp}) \quad \Rightarrow \quad \|\mathbf{y} - \mu_{\mathbf{Y}}\|^2 = \|\hat{\mathbf{y}}^* - \mathbf{y}\|^2 + \|\hat{\mathbf{y}}^* - \mu_{\mathbf{Y}}\|^2$$

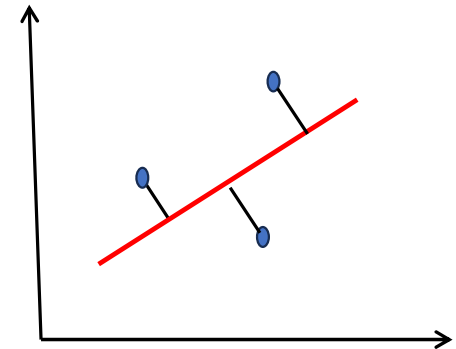
# Reconstruction Error

variance of data = reconstruction error + variance of reconstruction

$$\mathbb{E}[\|\mathbf{y} - \boldsymbol{\mu}_Y\|^2] = \mathbb{E}[\|\hat{\mathbf{y}}^* - \mathbf{y}\|^2] + \mathbb{E}[\|\hat{\mathbf{y}}^* - \boldsymbol{\mu}_Y\|^2]$$

Minimizing the reconstruction error = Maximizing the variance of reconstruction

- The reconstruction error is
  - The average squared perpendicular distances from each data point to the subspace spanned by the selected principal components.
  - The total lost variance (i.e. out of space variance)



# Lagrange Multipliers

- Find extrema  $E(\mathbf{x})$ , where  $\mathbf{x} \in \mathbb{R}^d$  subject to  $g(\mathbf{x}) = 0$ .
- Turn the constrained optimization problem into an unconstrained optimization problem by creating a new objective function called Lagrangian:

$$L(\mathbf{x}, \lambda) = E(\mathbf{x}) + \lambda g(\mathbf{x})$$

- Extrema of the unconstrained objective  $L(\mathbf{x}, \lambda)$  are the extrema of the original constrained  $E(\mathbf{x})$

$$\begin{cases} \frac{\partial}{\partial \mathbf{x}} L(\mathbf{x}, \lambda) = \nabla E(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0 \\ \frac{\partial}{\partial \lambda} L(\mathbf{x}, \lambda) = g(\mathbf{x}) = 0 \end{cases}$$

# Maximizing the Variance

- Given observed data matrix  $\mathbf{Y} \in \mathbb{R}^{d \times N}$ , find a **unit vector  $\mathbf{w}$**  such that  $\text{Var}(\mathbf{w}^T \mathbf{Y}) = \mathbf{w}^T \mathbf{S} \mathbf{w}$  is maximized.
- We have one constraint in our optimization problem

$$\|\mathbf{w}\| = \mathbf{w}^T \mathbf{w} = 1$$


Why do we need to have this constraint?

- Using Lagrange multipliers, we define the Lagrangian as

$$L(\mathbf{w}, \lambda) = \mathbf{w}^T \mathbf{S} \mathbf{w} + \lambda(1 - \mathbf{w}^T \mathbf{w})$$

# Solving the Lagrangian

- Set the partial derivative of  $L$  w.r.t  $\mathbf{w}$  to zero

$$\frac{\partial L}{\partial \mathbf{w}} = 2\mathbf{S}\mathbf{w} - 2\lambda\mathbf{w} = \mathbf{0}$$


$$\mathbf{S}\mathbf{w} = \lambda\mathbf{w}$$

- $\lambda$  and  $\mathbf{w}$  are the eigenvalue and eigenvector of the sample covariance matrix  $\mathbf{S}$
- There are at most  $d$  eigenvalues and eigenvectors, which  $\mathbf{w}$  maximizes the objective function?



# Maximizing the Variance

- The largest eigenvalue is the maximum variance
- The eigenvector corresponding to the maximum eigenvalue is the first principal component

$$E(\mathbf{w}) = \mathbf{w}^T \mathbf{S} \mathbf{w} = \mathbf{w}^T \lambda \mathbf{w} = \lambda \mathbf{w}^T \mathbf{w} = \lambda \|\mathbf{w}\| = \lambda$$

- **S** is symmetric and semi-definitive by definition
  - Eigenvectors are orthogonal to each other
  - Eigenvalues are real and non-negative
  - With the constraint  $\|\mathbf{w}\| = 1$ , the eigenvectors are orthonormal.

# PCA Algorithm

1. Compute sample mean  $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_i$ , where  $\mathbf{y}_i \in \mathbb{R}^d$
2. Compute the sample covariance matrix  $\mathbf{S} = \frac{1}{N-1} (\mathbf{Y} - \boldsymbol{\mu}\mathbf{1})(\mathbf{Y} - \boldsymbol{\mu}\mathbf{1})^T$
3. Let  $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T = \mathbf{S}$  be the eigenvector decomposition of  $\mathbf{S}$ , where
  - $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d)$
  - $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_d]$  contains orthonormal eigenvectors,  $\mathbf{V}\mathbf{V}^T = \mathbf{I}_d$
4. Sort the eigenvalues from largest to smallest ( $\lambda_1 \geq \lambda_2 \dots \geq \lambda_d$ ) along with their corresponding eigenvectors
5. Let  $\mathbf{W} = [\mathbf{v}_1, \dots, \mathbf{v}_k]$  be the matrix containing the first  $k$  eigenvectors
6. Let  $\mathbf{X} = \mathbf{W}^T (\mathbf{Y} - \boldsymbol{\mu}\mathbf{1})$  (i.e.  $\mathbf{x}_i = \mathbf{W}^T (\mathbf{y}_i - \boldsymbol{\mu})$  for  $\forall i \in \{1, \dots, N\}$ )

# How to choose $k$

## 1. Plot $\lambda_j$ vs. $j$

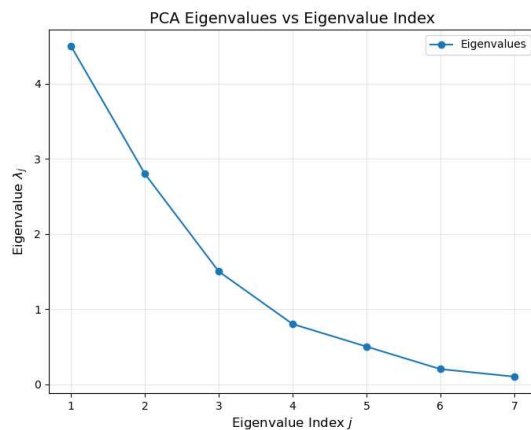
- The smaller  $\lambda_j$ , the less out of space variance depending how fast  $\lambda_j$  declines

## 2. Use the cumulative explained variance ratio $\rho$

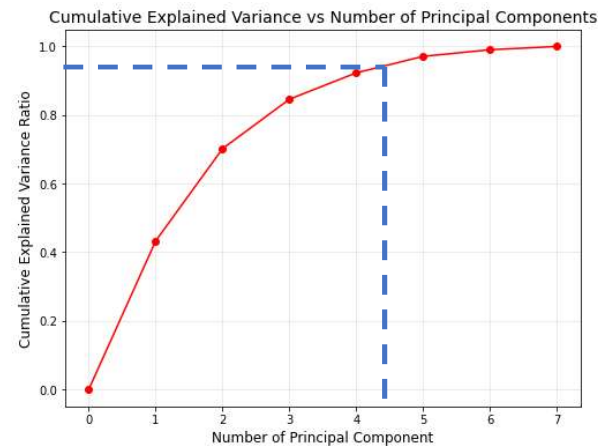
$$\rho = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^N \lambda_j} = \frac{\text{Variance in the Principal Component Subspace}}{\text{Total Variance of the Data}}$$

The **explained variance** is the amount of variability in the data captured by each principal component

1.



2.

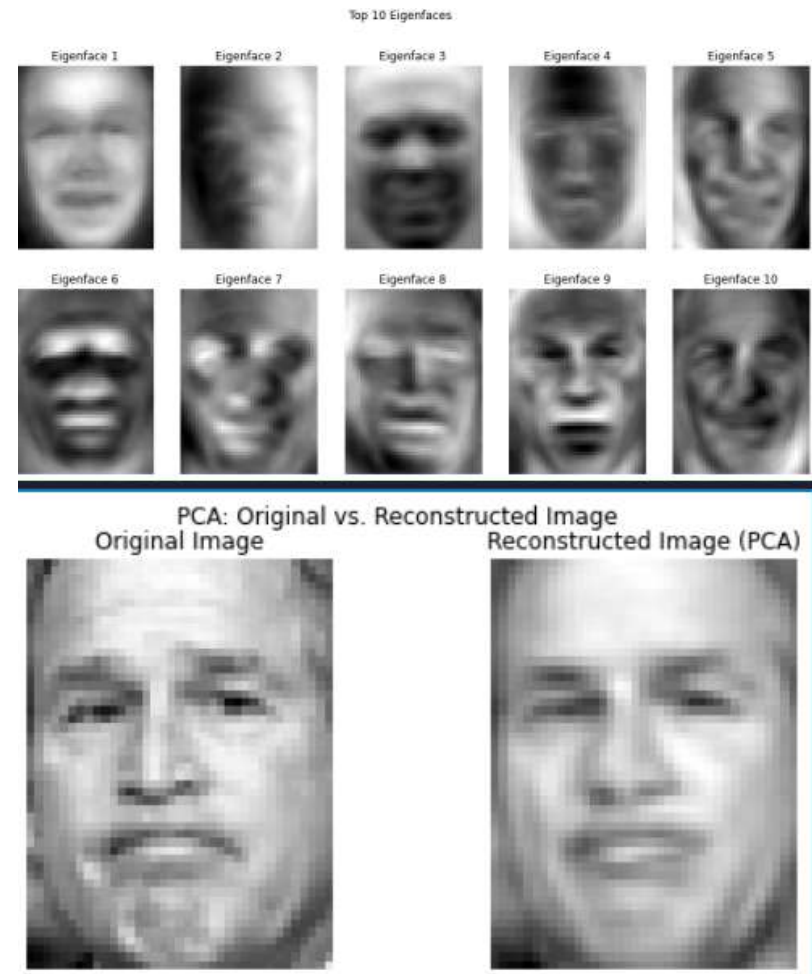


Choose 5

# Eigenfaces

Each picture is 50x37 pixels  
Flatten the image, we get 1850 column vector per image  
We have 1140 images  
Each principal component is a 1850X1 vector  
Reshape each vector back to 50 x37  
We got eigenfaces images  
We display top 10 eigenfaces

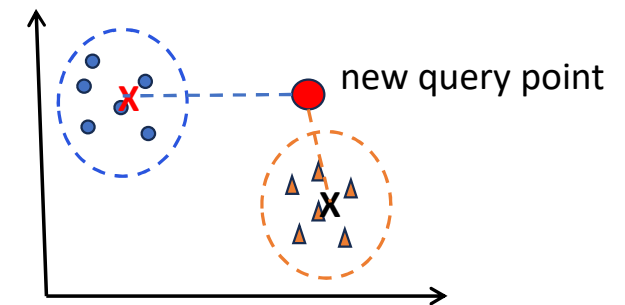
Pick one of the images and projected it to the  
subspace spanned by the 100 principal components.  
Then reconstruct the image



# Clustering

# Clustering

- A cluster is a collection of data points that share **similar** properties
- The underlying structure of data reflects some natural kinds
- Applications of clustering
  - Category discovery
    - Example: document categorization
    - Similarities
      - Spatial Layout
      - Words used in the documents
  - Vector Quantization
    - Use the center of the cluster to represent all points in the cluster
    - Store or transmit only the center of the data
  - Nearest neighbor search
    - Searching all cluster centers instead of all points



# K-Means

- Dataset  $\{\mathbf{y}_i\}_{i=1}^N, \mathbf{y}_i \in \mathbb{R}^d$
- $K$  distinct clusters (assume user defines it)
- Centers of cluster:  $\{\mathbf{c}_j\}_{j=1}^K, \mathbf{c}_j \in \mathbb{R}^d$ 
  - $\mathbf{c}_j$  is the cluster exemplar representing all points in the  $j^{\text{th}}$  cluster
  - Other names: centroid, prototype, mean
- Cluster assignment matrix (binary)  $\mathbf{L} \in \{0,1\}^{N \times K}$  with elements  $l_{i,j}$ 
  - 1-of- $K$  representation defines which point belongs to which cluster

$$l_{i,j} = \begin{cases} 1 & \text{if } \mathbf{y}_i \text{ is assigned to cluster } j \\ 0 & \text{otherwise} \end{cases}$$



$$\text{row sum} = 1$$

$$\sum_{j=1}^K l_{i,j} = 1$$

A point can only be assigned to one cluster

# K-Means

- $\{\mathbf{c}_j\}_{j=1}^K$  and  $\mathbf{L}$  are the unknowns we are looking for
- Goal: find groupings of data points (assignment matrix  $\mathbf{L}$ ) that are well represented by the cluster centers ( $\{\mathbf{c}_j\}_{j=1}^K$ )
- Objective function
  - How closely grouped data points are to their centers
  - **Inertia, WCSS** (within-cluster sum of squares)
  - A distortion measure

$$E(\mathbf{L}, \{\mathbf{c}_j\}_{j=1}^K) = \sum_{i=1}^N \sum_{j=1}^K l_{i,j} \|\mathbf{y}_i - \mathbf{c}_j\|^2$$

Only penalize when  $\mathbf{y}_i$  belongs to cluster  $j$

Squared Euclidean distance

A: sum over all clusters  
B: sum over all points



# About Assignment Matrix $\mathbf{L}$

$$C_1 = \{\mathbf{y}_1, \mathbf{y}_3\}$$

$$C_2 = \{\mathbf{y}_2, \mathbf{y}_5\}$$

$$C_3 = \{\mathbf{y}_4\}$$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Row sum of  $\mathbf{L}$

$$\sum_{j=1}^K l_{i,j} = 1$$

- Column sum of  $\mathbf{L}$

$$\sum_{i=1}^N l_{i,j} = \text{Number of points assigned to cluster } j$$

- Sum of all elements in  $\mathbf{L}$

$$\sum_{i=1}^N \sum_{j=1}^K l_{i,j} = \sum_{j=1}^K \sum_{i=1}^N l_{i,j} = N$$

- Sum of values of all points in Cluster  $j$

$$\sum_{i=1}^N l_{i,j} \mathbf{y}_i = \text{sum of } \mathbf{y}_i \text{ in cluster } j$$

# K-Means Algorithm (Lloyd's Algo.)

- Initialize  $\{\mathbf{c}_j\}_{j=1}^K$  and  $\mathbf{L}$
- Loop till max # iterations or convergence reached

1. Fix  $\{\mathbf{c}_j\}_{j=1}^K$ , update  $\mathbf{L} = \underset{\mathbf{L}}{\operatorname{argmin}} E(\mathbf{L}, \{\mathbf{c}_j\}_{j=1}^K)$

For each  $i \in \{1, \dots, N\}$ , find  $j^*$  such that

$$j^* = \underset{j}{\operatorname{argmin}} \|\mathbf{y}_i - \mathbf{c}_j\|^2 \quad \leftarrow \text{loop over } j?$$

$$l_{i,j^*} = 1$$

$$l_{i,j} = 0, \quad \forall j^*, j, \in \{1, \dots, K\} \text{ and } j \neq j^*$$

E-step: Update Labeling

2. Fix  $\mathbf{L}$ , update Centers  $\{\mathbf{c}_j\}_{j=1}^K$

$$\mathbf{c}_j = \underset{\mathbf{c}_j}{\operatorname{argmin}} E(\mathbf{L}, \{\mathbf{c}_j\}_{j=1}^K)$$

$$= \frac{\sum_{i=1}^N l_{i,j} \mathbf{y}_i}{\sum_{i=1}^N l_{i,j}}$$

$$= \frac{\text{sum of } \mathbf{y}_i \text{ in cluster } j}{\text{\# of points assigned to cluster } j}$$

M-step: Update Centers

Block coordinate descent

# Speedup Computation in Labeling

- **Precompute** and **Lookup** values to reduce computation in the loop

$$\|\mathbf{y}_i - \mathbf{c}_j\|^2 = (\mathbf{y}_i - \mathbf{c}_j)^T (\mathbf{y}_i - \mathbf{c}_j) = (\mathbf{y}_i^T \mathbf{y}_i - 2\mathbf{c}_j^T \mathbf{y}_i + \mathbf{c}_j^T \mathbf{c}_j)$$

$$\begin{array}{c} \text{precompute outside loop} \rightarrow \|\mathbf{y}_i\|^2 + \underbrace{\|\mathbf{c}_j\|^2}_{\text{precompute in each loop}} - 2\mathbf{c}_j^T \mathbf{y}_i \end{array}$$

- Matrix multiplication to compute cross term computation

$$\mathbf{Y} = \begin{bmatrix} | & | & | \\ \mathbf{y}_1 & \cdots & \mathbf{y}_N \\ | & | & | \end{bmatrix}_{d \times N} \quad \mathbf{C} = \begin{bmatrix} | & | & | \\ \mathbf{c}_1 & \cdots & \mathbf{c}_k \\ | & | & | \end{bmatrix}_{d \times K} \quad \mathbf{C}^T \mathbf{Y}$$

# About the Mean of the Cluster

Why  $\mathbf{c}_j = \underset{\mathbf{c}_j}{\operatorname{argmin}} E(\mathbf{L}, \mathbf{c}_j) = \frac{\sum_{i=1}^N l_{i,j} \mathbf{y}_i}{\sum_{i=1}^N l_{i,j}}$

$$\begin{aligned} E(\mathbf{L}, \mathbf{c}_j) &= \sum_{i=1}^N l_{i,j} \|\mathbf{y}_i - \mathbf{c}_j\|^2 \\ &= \sum_{i=1}^N l_{i,j} (\mathbf{y}_i - \mathbf{c}_j)^T (\mathbf{y}_i - \mathbf{c}_j) \\ &= \sum_{i=1}^N l_{i,j} (\mathbf{y}_i^T \mathbf{y}_i - 2\mathbf{y}_i^T \mathbf{c}_j + \mathbf{c}_j^T \mathbf{c}_j) \end{aligned}$$

$$\frac{\partial E}{\partial \mathbf{c}_j} = \sum_{i=1}^N l_{i,j} (-2\mathbf{y}_i + 2\mathbf{c}_j) = 0$$



$$\sum_{i=1}^N l_{i,j} \mathbf{y}_i = \sum_{i=1}^N l_{i,j} \mathbf{c}_j$$



$$\sum_{i=1}^N l_{i,j} \mathbf{y}_i = \mathbf{c}_j \sum_{i=1}^N l_{i,j}$$



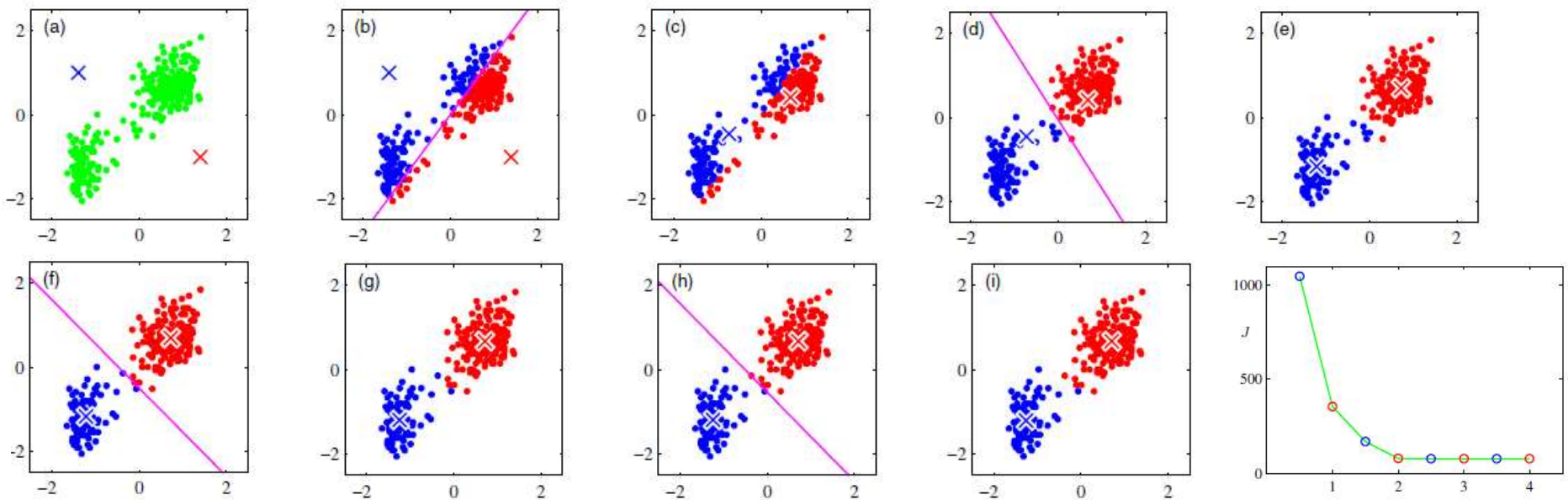
$$\mathbf{c}_j = \frac{\sum_{i=1}^N l_{i,j} \mathbf{y}_i}{\sum_{i=1}^N l_{i,j}}$$

# K-Means Convergence

- In the block coordinate descent,
  - We partition the variables to two subsets
  - We first hold the first subset of variables fixed and solve the closed form optimization problem of the second subset of variables
  - We then hold the second subset of variables fixed and solve the closed form optimization problem of the first subset of variables
  - We alternate between these two optimization problems
- Each optimization problem reduces the objective function value, and the objective function's lower bound is 0, hence the algorithm is guaranteed to converge.

# K-Means Convergence Cont'd

[Images Courtesy of Bishop's PRML]



**Figure 9.1** Illustration of the  $K$ -means algorithm using the re-scaled Old Faithful data set. (a) Green points denote the data set in a two-dimensional Euclidean space. The initial choices for centres  $\mu_1$  and  $\mu_2$  are shown by the red and blue crosses, respectively. (b) In the initial E step, each data point is assigned either to the red cluster or to the blue cluster, according to which cluster centre is nearer. This is equivalent to classifying the points according to which side of the perpendicular bisector of the two cluster centres, shown by the magenta line, they lie on. (c) In the subsequent M step, each cluster centre is re-computed to be the mean of the points assigned to the corresponding cluster. (d)–(i) show successive E and M steps through to final convergence of the algorithm.

**Figure 9.2** Plot of the cost function  $J$  given by (9.1) after each E step (blue points) and M step (red points) of the  $K$ -means algorithm for the example shown in Figure 9.1. The algorithm has converged after the third M step, and the final EM cycle produces no changes in either the assignments or the prototype vectors.

# Image Segmentation



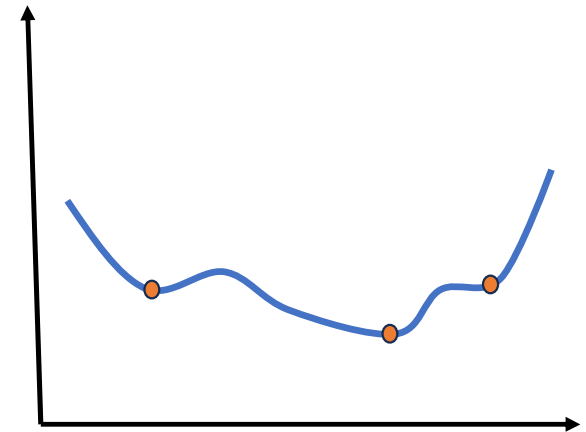
**Figure 9.3** Two examples of the application of the  $K$ -means clustering algorithm to image segmentation showing the initial images together with their  $K$ -means segmentations obtained using various values of  $K$ . This also illustrates the use of vector quantization for data compression, in which smaller values of  $K$  give higher compression at the expense of poorer image quality.

- Image segmentation Partitions an image into regions
  - Each region has a reasonably homogeneous visual appearance
  - Each region corresponds to objects or parts of objects.
- Vector quantization
  - Use the cluster mean to replace all points (i.e. pixels) in a cluster

[Images Courtesy of Bishop's PRML]

# K-Means Convergence Issues

- Non-convex optimization problem
- Converges to local optimum
- No guarantee to converge to global optimum
- K-Means is sensitive to initialization
- Poor initialization can lead to poor results





# K-Means Initialization

- Random  $\mathbf{L}$ , not very good
  - Every cluster owns a random subset of the whole dataset
  - Every mean is very close to the mean of the entire dataset
  - All initial centers are competing for the same point.
  - Slow, giving suboptimal solution
- Random  $\{\mathbf{c}_j\}_{j=1}^K$ 
  - If data are evenly distributed, it might be OK
  - If data occupies certain regions of space, it may assign centers to places with no data points
- Randomly choose data points as cluster centers
  - Fast initialization, work well in a good fraction of time
  - Often falls into local optimum

# K-Means Initialization

- Multiple restarts
  - Run the algorithm 10 -30 times
  - Each time choose random data points to initialize centers
  - Choose the solution with the smallest value of the objective function
- K-Means++
  - Choose your cluster centers so that they are
    - close to the data points
    - very spread out (i.e. clusters are sufficiently far away and being diverse)

# K-Means++

1. Choose  $\mathbf{c}_1$  at random from  $\{\mathbf{y}_i\}_{i=1}^N$
2. Choose the next  $\mathbf{c}_2$  reasonably far from  $\mathbf{c}_1$ 
  - The one that is the furthest from  $\mathbf{c}_1$  might be far away from all other points
  - We should select the next point that is reasonably far but not too far away
  - $\forall \mathbf{y}_i$ , let  $d_i$  be the distance between  $\mathbf{y}_i$  and the closest cluster found so far
  - Let

$$p_i = \frac{d_i^2}{\sum_{i=1}^N d_i^2}$$

← The further away the data, the more likely to be chosen

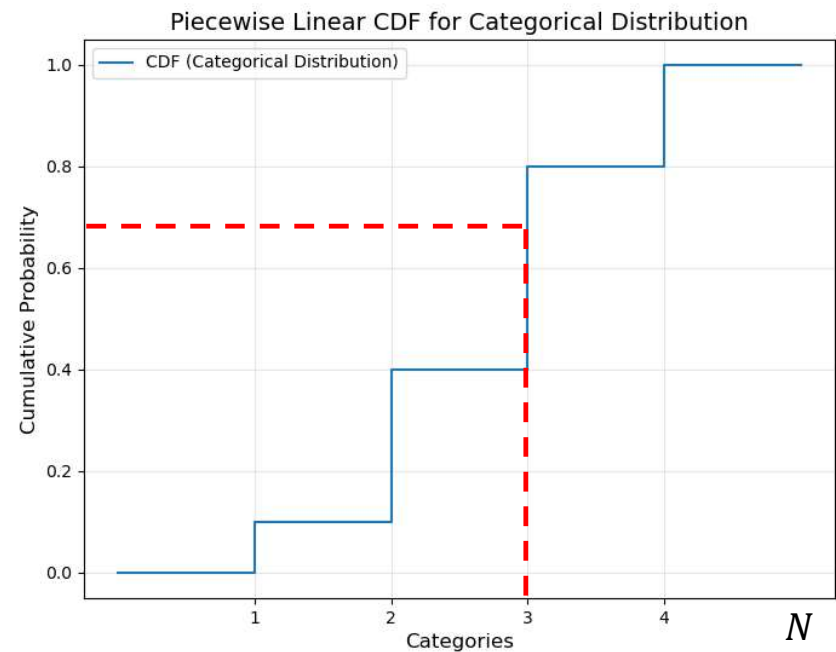
← Normalization term

- Choose next cluster center using  $p_i$ , which is a categorical probability distribution for a large dataset (i.e.  $i = \{1, \dots, N\}$ )

# Sampling

- Sample from  $\{\mathbf{y}_i\}_{i=1}^N$  with probability  $p_i$
- Refer to Tutorial 10 and lecture notes Section 13.2
- Sample  $Z \sim U[0,1]$
- Use index given by  $\lfloor \text{CDF}^{-1}(Z) \rfloor$

$$\text{CDF}(j) = \sum_{i=1}^j p_i$$

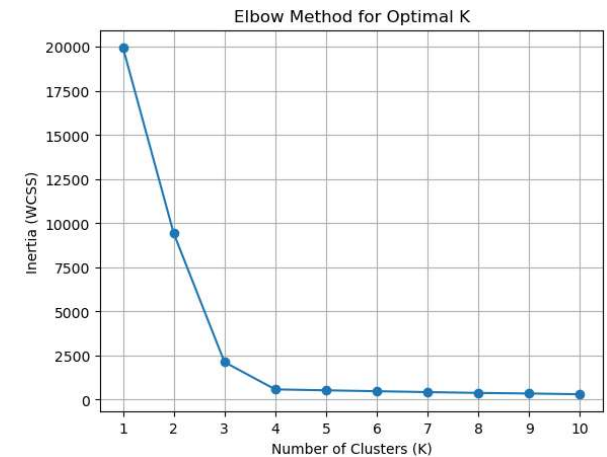


# Choose $K$ : Non-trivial

- Use the training set
  - Compute the inertia for different  $K$ , the Within-Cluster Sum of Squares (WCSS)

$$\text{WCSS} = \sum_{i=1}^N \sum_{j=1}^K l_{i,j} \|\mathbf{y}_i - \mathbf{c}_j\|^2$$

- Use the elbow method
- Use a holdout validation set
  - Train the data on the training set
  - Compute performance metric on the validation set
  - Check how consistent the performance metric across the two sets of data



# Choose $K$ Cont'd

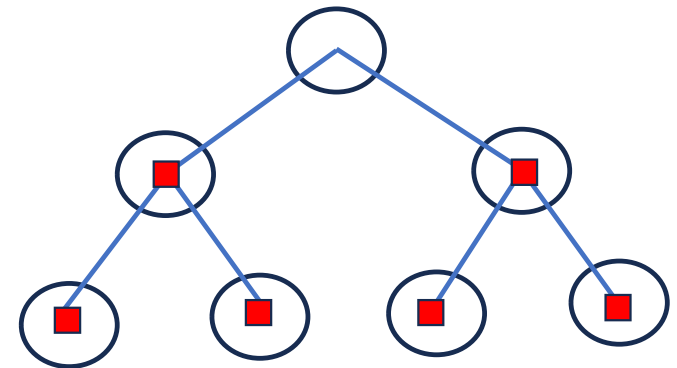
- Penalized likelihood
  - Like Bayesian model selection
  - Allows  $K$  to grow as large as we want
  - Add a penalty onto the objective function which penalizes the number of clusters
- Latent Dirichlet Analysis (LDA)
  - More rigorous formal method (not in the scope of this course)

# Variants of K-Means

- K-Medoids
  - Not using the mean to represent the cluster
  - Use the point closest to the mean (i.e. center) to represent the cluster
- Hierarchical K-Means
  - Given a K-Means solution, how do we find the closest cluster for a given data point that we want to query about?
  - Vanilla K-Means algorithm requires searching K clusters.
  - When K is large, storing and searching K clusters becomes very expensive.
  - Solution: build cluster centers in a hierarchical fashion

# Hierarchical K-Means Cont'd

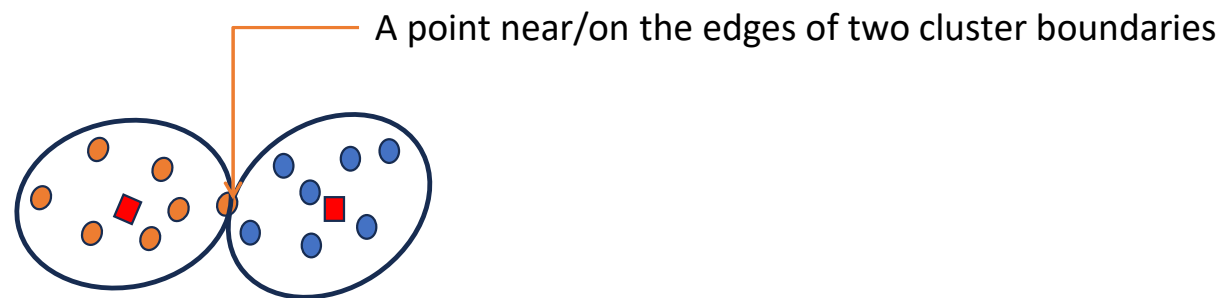
- Build cluster centroids in a hierarchical fashion
- Start with the whole dataset
- Break it to  $k_i$  (i.e.  $k_i = 2$ ) clusters at each level  $i$
- The number of clusters grow exponentially
- $K$  cluster in total,  $\log K$  levels
  - $K$  is in the 100K – 1M magnitude
  - It greatly helps the query speed
- Approximate Nearest Neighbor (ANN)
  - Build multiple paths for close enough clusters
  - Find multiple clusters and find NN within those clusters
  - Specify approximation criteria to bound accuracy
    - How many neighbors you might miss





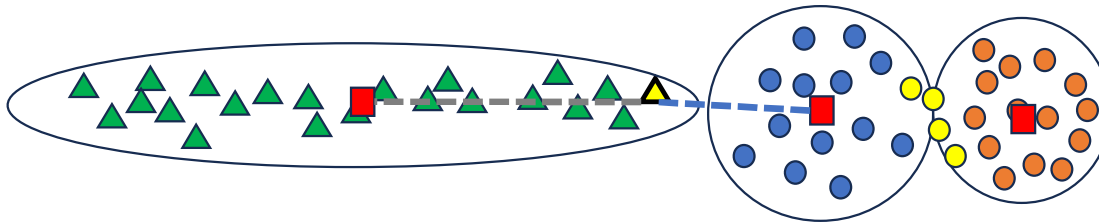
# Variants of K-Means Cont'd

- Probabilistic Approach
- Represent the cluster by the probability distribution of the data
  - Mean + Variance
- Mixture model
  - Each point belongs to multiple probability distributions



# Mixture Models

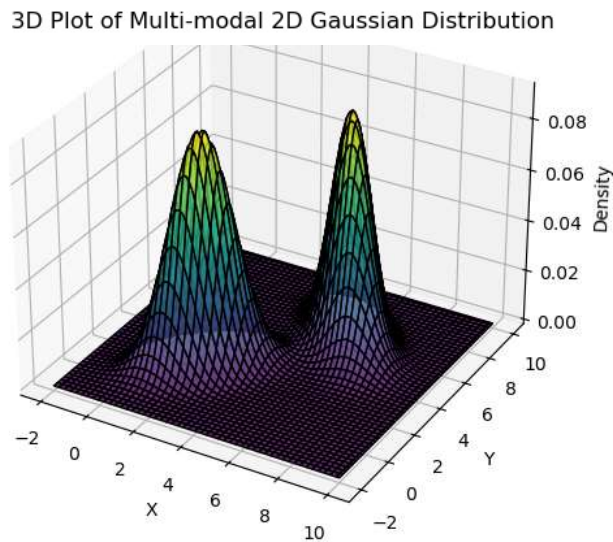
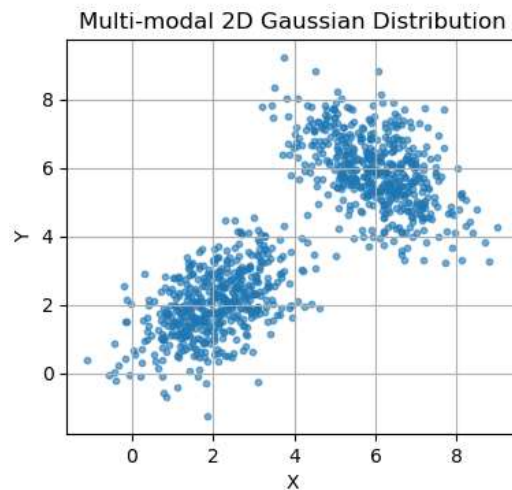
# Probabilistic Mixture Model



- Data distributions that are not good for K-Means
  - Points that are near the boundary
  - Points are from an elongated data distribution
- K-Means is a hard assignment.
  - A point is assigned to a cluster with 0 or 1.
  - It is good for spherical, well-separated points
- Probabilistic Mixture Model is a soft assignment.
  - We assign a point to a cluster with a probability that the cluster generates it

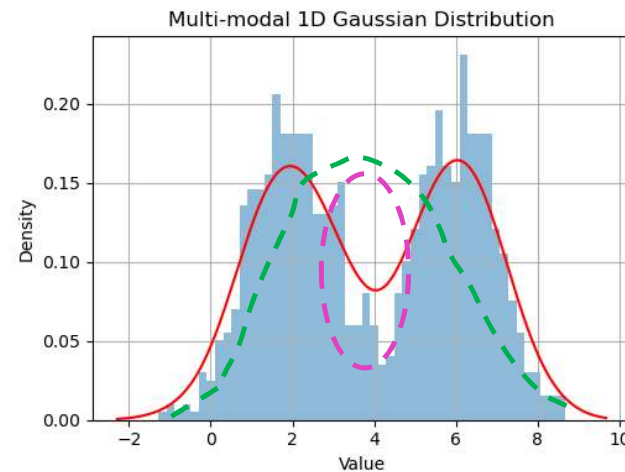
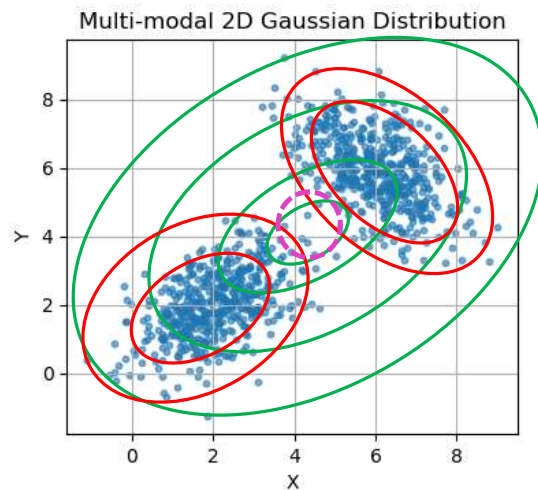
# Multimodal Distribution

- Example of two clusters, each contains data from a 2D Gaussian Distribution.
- There are two modes in data distribution



# Multimodal Distribution

- When the data distribution has multiple modes, using a single mode data distribution will incorrectly assign high probability to regions we have very small amount of data or even no data.



The green density function gives high density to areas with small amount of data

# Mixture Model

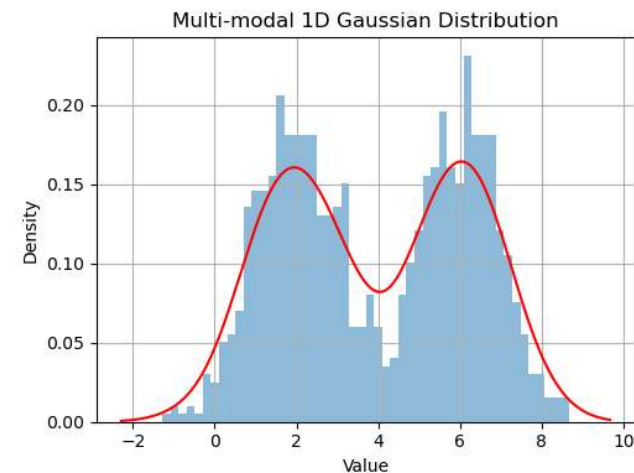
- Model the probability distribution as a weighted sum of multiple (single mode) data distributions

$$p(\mathbf{y}) = \sum_{j=1}^K m_j p(\mathbf{y}; \theta_j)$$

$$\begin{cases} m_j \in [0,1] \\ \sum_{j=1}^K m_j = 1 \end{cases}$$

- $m_j$  is the weight:
  - It is the mixing probability, which has a categorical distribution
- $p(\mathbf{y}; \theta_j)$  is the component density function
- $\theta_j$  is the parameters of the  $j^{\text{th}}$  density function

This is similar to basis function linear regression, but with the constraints that weights are non-negative and sum to 1 and each basis function is a probability density function



# Gaussian Mixture Model

- The mixture model allows any types of distribution mixing
- We will focus on Gaussian Mixture Model
- We start with 1D case:  $\{y_i\}_{i=1}^N, y_i \in \mathbb{R}$
- Case 1:  $K = 1$ , one component, parameters are

$$m_1 = 1 \quad \mu = \frac{1}{N} \sum_{i=1}^N y_i \quad \sigma^2 = \frac{1}{N} \sum_{i=1}^N (y_i - \mu)^2$$

We use population variance here  
You can also use sample variance

# Gaussian Mixture Model

Case 2:  $K > 1$ , multiple components,

- Assume we know cluster assignment variable  $l_{i,j}$

$$l_{i,j} = \begin{cases} 1 & \text{if } y_i \text{ is assigned to cluster } j \\ 0 & \text{otherwise} \end{cases}$$

- We will partition the data to  $K$  components and compute the mean and variance for each component

$$m_j = \frac{\sum_{i=1}^N l_{i,j}}{N} \quad \mu_j = \frac{\sum_{i=1}^N (l_{i,j} y_i)}{\sum_{i=1}^N l_{i,j}} \quad \sigma_j^2 = \frac{\sum_{i=1}^N l_{i,j} (y_i - \mu_j)^2}{\sum_{i=1}^N l_{i,j}}$$



# Probabilistic Mixture Model

- In  $K$ -Means, we determine  $l_{i,j}$  using the distance between the point and its closest centroid.
- In Probabilistic Mixture Model, we use the posterior probabilistic distribution over the assignment variable  $l_{i,j}$

$$p(l = j|y_i) = \frac{\overset{\text{Prior}}{p(l = j)}\overset{\text{Likelihood}}{p(y_i|l = j)}}{p(y_i)}$$

Posterior about assigning data to clusters

Marginal probability of data

# Example: Two Component GMM

- Posterior over cluster assignment variable given data and the model parameters is Gaussian

$$p(l = 1|y_i, \theta) \quad p(l = 2|y_i, \theta)$$

- Model parameters are

$$\theta = \{\mu_1, \mu_2, \sigma_1^2, \sigma_2^2\}$$

- Assume equal Prior on the two Gaussians

$$p(l = 1) = p(l = 2) = 1/2$$

## Example: Two Component GMM Cont'd

- Assume Gaussian Likelihood

$$p(y|l = 1, \theta) = G(y; \mu_1, \sigma_1^2) \quad p(y|l = 2, \theta) = G(y; \mu_2, \sigma_2^2)$$

- **Posterior (ownership) probability:**  $\gamma_{i,j} \equiv p(l = j|y_i, \theta)$

- AKA **responsibility** (cluster  $j$  is responsible for data point  $i$ )

$$\gamma_{i,1} = \frac{p(l = 1)p(y_i|l = 1, \mu_1, \sigma_1^2)}{p(l = 1)p(y_i|l = 1, \mu_1, \sigma_1^2) + p(l = 2)p(y_i|l = 2, \mu_2, \sigma_2^2)}$$

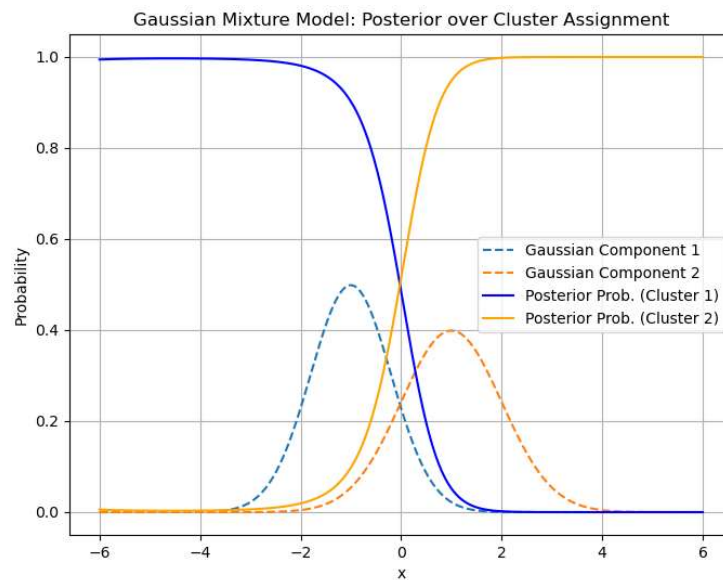
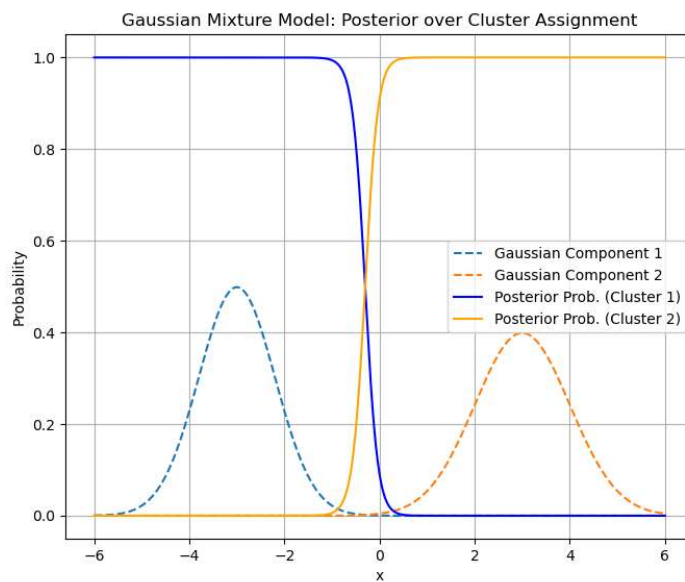
$$= \frac{1/2 G(y; \mu_1, \sigma_1^2)}{1/2(G(y; \mu_1, \sigma_1^2) + G(y; \mu_2, \sigma_2^2))}$$

$$p(l = 1) = p(l = 2) = 1/2$$

$$\gamma_{i,2} = 1 - \gamma_{i,1}$$

$$\gamma_{i,1} = \frac{1/2 G(y; \mu_1, \sigma_1^2)}{1/2(G(y; \mu_1, \sigma_1^2) + G(y; \mu_2, \sigma_2^2))}$$

$$\gamma_{i,2} = 1 - \gamma_{i,1}$$



$\gamma_{i,j}$  are soft weights

$$\mu_j = \frac{\sum_{i=1}^N (\gamma_{i,j} y_i)}{\sum_{i=1}^N \gamma_{i,j}}$$

$$\sigma_j^2 = \frac{\sum_{i=1}^N \gamma_{i,j} (y_i - \mu_j)^2}{\sum_{i=1}^N \gamma_{i,j}}$$

# General Formulation of GMM

- Dataset  $\{\mathbf{y}_i\}_{i=1}^N, \mathbf{y}_i \in \mathbb{R}^d$
- $K$  Gaussian component densities, each with  $\boldsymbol{\mu}_j \in \mathbb{R}^d, \mathbf{C}_j \in \mathbb{R}^{d \times d}$ 
  - $\boldsymbol{\mu}_{1:K}$  :  $K$  mean vectors
  - $\mathbf{C}_{1:K}$  :  $K$  covariance matrices
- Prior
  - $m_{1:K}$  : mixing probabilities  $\begin{cases} m_j \in [0,1] \\ \sum_{j=1}^K m_j = 1 \end{cases}$ 
    - $m_j$  is the fraction of data we expect to be generated by the  $j^{\text{th}}$  Gaussian Component
- Parameters
  - $\theta = \{m_{1:K}, \boldsymbol{\mu}_{1:K}, \mathbf{C}_{1:K}\}$

- Component Likelihood: Gaussian

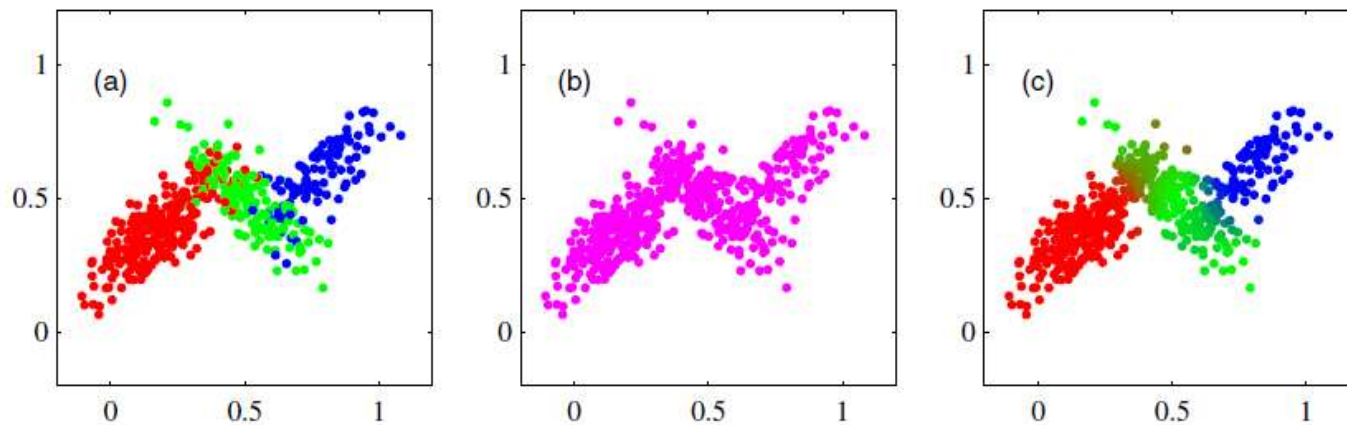
$$p(\mathbf{y}|l = j, \theta) = G(\mathbf{y}; \boldsymbol{\mu}_j, \mathbf{C}_j)$$

- Posterior (responsibilities)

$$\gamma_{i,j} \equiv p(l = j|\mathbf{y}_i, \theta)$$

$$\begin{aligned} &= \frac{p(l = j|\theta)p(\mathbf{y}_i|l = j, \theta)}{p(\mathbf{y}_i|\theta)} \\ &= \frac{m_j p(\mathbf{y}_i|l = j, \theta)}{p(\mathbf{y}_i|\theta)} \end{aligned}$$

# Example: Mixture of Gaussian [CB]



**Figure 9.5** Example of 500 points drawn from the mixture of 3 Gaussians shown in Figure 2.23. (a) Samples from the joint distribution  $p(z)p(x|z)$  in which the three states of  $z$ , corresponding to the three components of the mixture, are depicted in red, green, and blue, and (b) the corresponding samples from the marginal distribution  $p(x)$ , which is obtained by simply ignoring the values of  $z$  and just plotting the  $x$  values. The data set in (a) is said to be *complete*, whereas that in (b) is *incomplete*. (c) The same samples in which the colours represent the value of the responsibilities  $\gamma(z_{nk})$  associated with data point  $x_n$ , obtained by plotting the corresponding point using proportions of red, blue, and green ink given by  $\gamma(z_{nk})$  for  $k = 1, 2, 3$ , respectively

# GMM Data Likelihood

$$p(\mathbf{y}|\theta) = \sum_{j=1}^k p(\mathbf{y}, l = j|\theta)$$

Marginalization over latent variable  $l$

$$= \sum_{j=1}^k p(l = j|\theta)p(\mathbf{y}|l = j, \theta)$$

prior  $\times$  conditional likelihood

$$= \sum_{j=1}^k m_j G(\mathbf{y}; \boldsymbol{\mu}_j, \mathbf{C}_j)$$

$$= \sum_{j=1}^k m_j \left( \frac{1}{(2\pi)^d |\mathbf{C}_j|} \right)^{1/2} \exp \left( -\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu}_j)^T \mathbf{C}_j^{-1} (\mathbf{y} - \boldsymbol{\mu}_j) \right)$$

# GMM Generative Model

$$p(\mathbf{y}|\theta) = \sum_{j=1}^k m_j \left( \frac{1}{(2\pi)^d |\mathbf{C}_j|} \right)^{1/2} \exp \left( -\frac{1}{2} (\mathbf{y} - \boldsymbol{\mu}_j)^T \mathbf{C}_j^{-1} (\mathbf{y} - \boldsymbol{\mu}_j) \right)$$

- Draw  $l$  ( $l \in \{1, \dots, K\}$ ) from categorical distribution with probability  $m_j$
- Given  $l$ , draw an observation from component  $l$

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_l, \mathbf{C}_l)$$

- Note here we focus on Gaussian mixture model. In general, you could have different mixed distributions, where each component can be different family of distributions as well.
- $p(\mathbf{y}|\theta)$  is a weighted sum of Gaussians, similar to RBF regression



# Learning GMM

- Goal: Find  $\theta$  of GMM to maximize the data likelihood or equivalently to minimize the negative log likelihood of data
- Objective function:

$$L(\theta) = -\log p(\mathbf{y}_{1:N}|\theta)$$

- Key differences between RBF regression and GMM

Model	RBF Regression	GMM
Weight Constraints	None	$\begin{cases} m_j \in [0,1] \\ \sum_{j=1}^K m_j = 1 \end{cases}$
Loss Function	Squared error	Negative log loss of data likelihood

# GMM: ML Estimate

- Assume data points  $\mathbf{y}_{1:N}$  are i.i.D

$$L(\theta) = -\log p(\mathbf{y}_{1:N}|\theta) = -\log \prod_{i=1}^N p(\mathbf{y}_i|\theta)$$

$$= -\sum_{i=1}^N \log \sum_{j=1}^K m_j p(\mathbf{y}_i|l=j, \theta)$$

$$\theta = \{m_{1:K}, \boldsymbol{\mu}_{1:K}, \mathbf{C}_{1:K}\}$$

- Constraints

$$1. \begin{cases} m_j \in [0,1] \\ \sum_{j=1}^K m_j = 1 \end{cases}$$

2.  $\mathbf{C}_j$  is symmetric positive definite

# EM Algorithm

- Initialize  $\gamma_{i,j}$  and  $\theta$
- Loop till max # iterations or convergence reached

## 1. E-Step, fix $\theta$ , update $\gamma_{i,j}$

For each  $i \in \{1, \dots, N\}$ ,

$$\begin{aligned}\gamma_{i,j} &= p(l = j | \mathbf{y}_i, \theta) \\ &= \frac{p(l = j | \theta) p(\mathbf{y}_i | l = j, \theta)}{p(\mathbf{y}_i | \theta)} \\ &= \frac{m_j p(\mathbf{y}_i | l = j, \theta)}{\sum_{h=1}^K m_h p(\mathbf{y}_i | l = h, \theta)}\end{aligned}$$

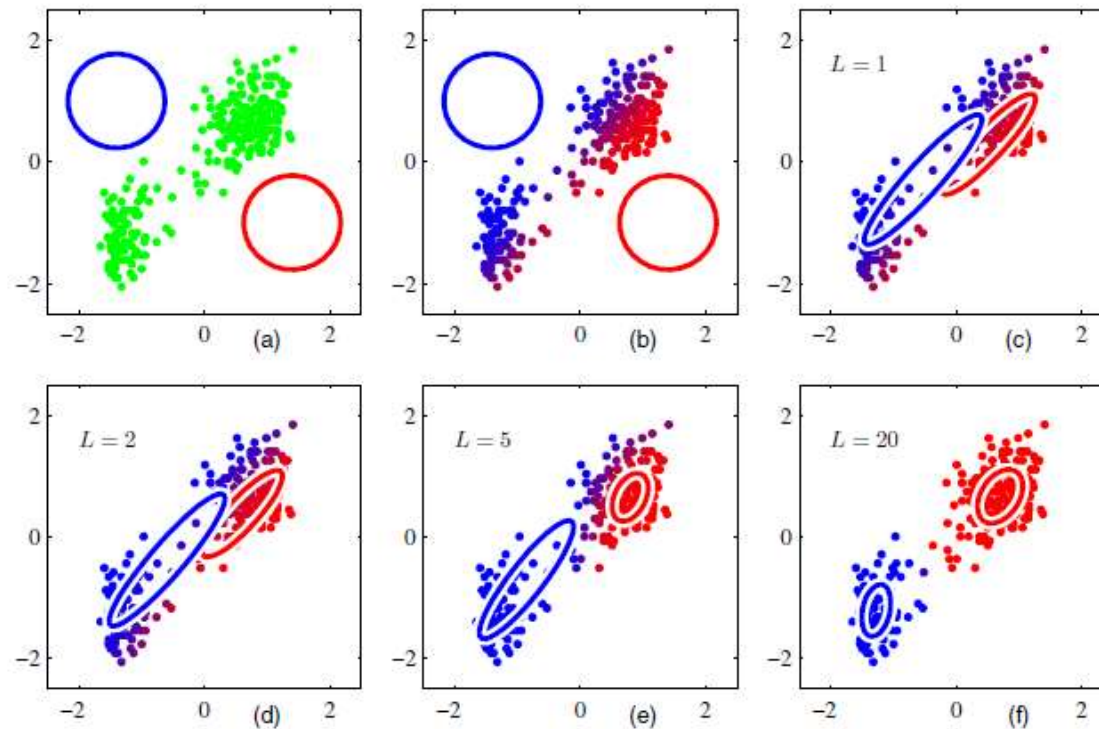


## 2. M-step, fix $\gamma_{i,j}$ , update $\theta$

For each cluster  $j$

$$\begin{aligned}m_j &= \frac{\sum_i \gamma_{i,j}}{N} \\ \boldsymbol{\mu}_j &= \frac{\sum_i (\gamma_{i,j} \mathbf{y}_i)}{\sum_i \gamma_{i,j}} \\ \mathbf{C}_j &= \frac{\sum_i \gamma_{i,j} (\mathbf{y}_i - \boldsymbol{\mu}_j)(\mathbf{y}_i - \boldsymbol{\mu}_j)^T}{\sum_i \gamma_{i,j}}\end{aligned}$$

# Example: EM Algorithm [CB]



**Figure 9.8** Illustration of the EM algorithm using the Old Faithful set as used for the illustration of the  $K$ -means algorithm in Figure 9.1. See the text for details.

# EM Algorithm Derivation

Parameter	Details	Description
$m_{1:K}$	$\sum_{j=1}^K m_j = 1$	Mixing probabilities
$\psi_{1:K}$	$\psi_j = \{\boldsymbol{\mu}_j, \mathbf{C}_j\}$	Likelihood parameters, Gaussian
$\theta$	$\{m_{1:K}, \psi_{1:K}\}$	Model parameters

- Lagrangian Function

$$\begin{aligned}
 L(\theta, \lambda) &= -\sum_{i=1}^N \log \sum_{j=1}^K m_j p(\mathbf{y}_i | l = j, \boldsymbol{\theta}) + \lambda (\sum_{j=1}^K m_j - 1) \\
 &= -\sum_{i=1}^N \log \sum_{j=1}^K m_j p(\mathbf{y}_i | \boldsymbol{\psi}_j) + \lambda (\sum_{j=1}^K m_j - 1)
 \end{aligned}$$

See p65

- Why no constraints of  $m_j \in [0,1]$  and  $\mathbf{C}_j$  being symmetric positive definite?

# EM Algorithm Derivation

$$L(\theta, \lambda) = -\sum_{i=1}^N \log \sum_{j=1}^K m_j p(\mathbf{y}_i | \psi_j) + \lambda (\sum_{j=1}^K m_j - 1)$$

- Necessary conditions for  $L(\theta)$  to reach optimum value are

$$\frac{\partial L}{\partial \lambda} = 0 \quad \Rightarrow \quad \sum_{j=1}^K m_j - 1 = 0 \quad \Rightarrow \quad \sum_{j=1}^K m_j = 1$$

$$\frac{\partial L}{\partial m_j} = 0$$

$$\frac{\partial L}{\partial \psi_j} = 0$$

About  $\frac{\partial L}{\partial m_j}$

$$L(\theta, \lambda) = -\sum_{i=1}^N \log \sum_{j=1}^K m_j p(\mathbf{y}_i | \psi_j) + \lambda \sum_{j=1}^K (m_j - 1)$$

$$\begin{aligned} \frac{\partial L}{\partial m_j} &= -\sum_{i=1}^N \frac{1}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \frac{\partial}{\partial m_j} \sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h) + \lambda \\ &= -\sum_{i=1}^N \frac{1}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \frac{\partial}{\partial m_j} [m_j p(\mathbf{y}_i | \psi_j) + \sum_{h \neq j} m_h p(\mathbf{y}_i | \psi_h)] + \lambda \\ &= -\sum_{i=1}^N \frac{1}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \frac{\partial}{\partial m_j} [m_j p(\mathbf{y}_i | \psi_j)] + \lambda \\ &= -\sum_{i=1}^N \frac{1}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} p(\mathbf{y}_i | \psi_j) + \lambda \end{aligned}$$

$m_j$  and  $\gamma_{i,j}$

$$\frac{\partial L}{\partial m_j} = -\sum_{i=1}^N \frac{1}{\sum_{h=1}^K m_h p(\mathbf{y}_i|\psi_h)} p(\mathbf{y}_i|\psi_j) + \lambda$$

See pg61

$$\gamma_{i,j} = \frac{m_j p(\mathbf{y}_i|\psi_j)}{\sum_{h=1}^K m_h p(\mathbf{y}_i|\psi_h)}$$



$$\frac{\gamma_{i,j}}{m_j} = \frac{p(\mathbf{y}_i|\psi_j)}{\sum_{h=1}^K m_h p(\mathbf{y}_i|\psi_h)}$$



$$m_j = \frac{1}{\lambda} \sum_{i=1}^N \gamma_{i,j}$$



$$\frac{\partial L}{\partial m_j} = -\sum_{i=1}^N \frac{\gamma_{i,j}}{m_j} + \lambda = 0$$



## $m_j$ and $\gamma_{i,j}$ Cont'd

$$\begin{cases} m_j = \frac{1}{\lambda} \sum_{i=1}^N \gamma_{i,j} \\ \sum_{j=1}^K m_j = 1 \end{cases} \quad \Rightarrow \quad \sum_{j=1}^K \frac{1}{\lambda} \sum_{i=1}^N \gamma_{i,j} = 1 \quad \Rightarrow \quad \frac{1}{\lambda} \sum_{i=1}^N \sum_{j=1}^K \gamma_{i,j} = 1$$

$$\sum_{j=1}^K \gamma_{i,j} = 1 \quad \Rightarrow \quad \frac{1}{\lambda} \sum_{i=1}^N 1 = 1 \quad \Rightarrow \quad \lambda = N$$

$$m_j = \frac{1}{N} \sum_{i=1}^N \gamma_{i,j}$$

About  $\frac{\partial L}{\partial \psi_j}$

$$L(\theta, \lambda) = -\sum_{i=1}^N \log \sum_{j=1}^K m_j p(\mathbf{y}_i | \psi_j) + \lambda \sum_{j=1}^K (m_j - 1)$$

$$\begin{aligned} \frac{\partial L}{\partial \psi_j} &= -\sum_{i=1}^N \frac{1}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \frac{\partial}{\partial \psi_j} \sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h) \\ &= -\sum_{i=1}^N \frac{1}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \frac{\partial}{\partial \psi_j} [m_j p(\mathbf{y}_i | \psi_j) + \sum_{h \neq j} m_h p(\mathbf{y}_i | \psi_h)] \\ &= -\sum_{i=1}^N \frac{1}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \frac{\partial}{\partial \psi_j} [m_j p(\mathbf{y}_i | \psi_j)] \\ &= -\sum_{i=1}^N \frac{m_j}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \frac{\partial}{\partial \psi_j} p(\mathbf{y}_i | \psi_j) \end{aligned}$$

# About $\frac{\partial L}{\partial \psi_j}$ Cont'd

$$\frac{\partial L}{\partial \psi_j} = - \sum_{i=1}^N \frac{m_j}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \frac{\partial}{\partial \psi_j} p(\mathbf{y}_i | \psi_j)$$

- We want to connect  $\frac{\partial p(\mathbf{y}_i | \psi_j)}{\partial \psi_j}$  with  $\log p(\mathbf{y}_i | \psi_j)$  because we will get a quadratic term

$$\frac{\partial}{\partial \psi} \log p(\psi) = \frac{1}{p(\psi)} \frac{\partial}{\partial \psi} p(\psi) \quad \Rightarrow \quad p(\psi) \frac{\partial}{\partial \psi} \log p(\psi) = \frac{\partial}{\partial \psi} p(\psi)$$

$$\frac{\partial L}{\partial \psi_j} = - \sum_{i=1}^N \left( \frac{m_j p(\mathbf{y}_i | \psi_j)}{\sum_{h=1}^K m_h p(\mathbf{y}_i | \psi_h)} \right) \frac{\partial}{\partial \psi_j} \log p(\mathbf{y}_i | \psi_j)$$

$$= - \sum_{i=1}^N \gamma_{i,j} \frac{\partial}{\partial \psi_j} \log p(\mathbf{y}_i | \psi_j)$$

Weighted sum of log likelihood

$$\frac{\partial}{\partial \psi_j} \log p(\mathbf{y}_i | \psi_j)$$

$$p(\mathbf{y}_i | \psi_j) = \frac{1}{(2\pi)^{d/2} |\mathbf{C}_j|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu}_j)^T \mathbf{C}_j^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_j)\right)$$

$$-\log p(\mathbf{y}_i | \psi_j) = \frac{d}{2} \log 2\pi + \frac{1}{2} \log |\mathbf{C}_j| + \frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu}_j)^T \mathbf{C}_j^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_j)$$

$$-\frac{\partial}{\partial \boldsymbol{\mu}_j} \log p(\mathbf{y}_i | \boldsymbol{\mu}_j, \mathbf{C}_j) = \mathbf{C}_j^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_j)$$

$$-\frac{\partial}{\partial \mathbf{C}_j^{-1}} \log p(\mathbf{y}_i | \boldsymbol{\mu}_j, \mathbf{C}_j) = \frac{1}{2} (\mathbf{y}_i - \boldsymbol{\mu}_j)(\mathbf{y}_i - \boldsymbol{\mu}_j)^T - \frac{1}{2} \mathbf{C}_j$$

See Tutorial 05

## Getting $\boldsymbol{\mu}_j$ and $\mathbf{C}_j$

$$\frac{\partial L}{\partial \psi_j} = -\sum_{i=1}^N \gamma_{i,j} \frac{\partial}{\partial \psi_j} \log p(\mathbf{y}_i | \psi_j)$$

$$\frac{\partial L}{\partial \boldsymbol{\mu}_j} = -\sum_{i=1}^N \gamma_{i,j} \mathbf{C}_j^{-1} (\mathbf{y}_i - \boldsymbol{\mu}_j) = 0$$



$$\boldsymbol{\mu}_j = \frac{\sum_i (\gamma_{i,j} \mathbf{y}_i)}{\sum_i \gamma_{i,j}}$$

$$\frac{\partial}{\partial \mathbf{C}_j^{-1}} \log p(\mathbf{y}_i | \boldsymbol{\mu}_j) = \frac{1}{2} \sum_{i=1}^N \gamma_{i,j} \left( (\mathbf{y}_i - \boldsymbol{\mu}_j)(\mathbf{y}_i - \boldsymbol{\mu}_j)^T - \mathbf{C}_j \right)$$

$$= \frac{1}{2} \sum_{i=1}^N \gamma_{i,j} (\mathbf{y}_i - \boldsymbol{\mu}_j)(\mathbf{y}_i - \boldsymbol{\mu}_j)^T - \frac{1}{2} \mathbf{C}_j \sum_{i=1}^N \gamma_{i,j} = 0$$



$$\frac{\partial}{\partial \mathbf{C}_j^{-1}} f(\mathbf{C}_j) = 0 \rightarrow \frac{\partial}{\partial \mathbf{C}_j} f(\mathbf{C}_j) = 0$$

$$\mathbf{C}_j = \frac{\sum_i \gamma_{i,j} (\mathbf{y}_i - \boldsymbol{\mu}_j)(\mathbf{y}_i - \boldsymbol{\mu}_j)^T}{\sum_i \gamma_{i,j}}$$

# Relation to K-Means

EM reduces to K-Means when

- Mixing probabilities are equal:  $m_j = 1$ 
  - No adjustment of mixture weights
- The Gaussians are spherical with identical variances
  - $\forall j, \mathbf{C}_j = \sigma^2 \mathbf{I}$
  - No flexibility in modeling clusters with different shapes
- The Gaussian variances are infinitesimal (i.e.  $\sigma^2 \rightarrow 0$ )
  - $\lim_{\sigma^2 \rightarrow 0} P(l = j | \mathbf{y}_i) = 1$
  - Soft membership assignment  $\gamma_{i,j}$  becomes binary hard assignment

# Acknowledgement

- Prof. David Fleet developed the course. He made his notes and courseware available to all of us.
- Prof. Francisco (Paco) Estrada shared his assignments and insights.
- Prof. Rawad A. Assi shared past assignments and advices.