MAXIMUM LIKELIHOOD ESTIMATION

DATA/MSML 603: Principles of Machine Learning

Shortcoming of BDT

- Bayesian decision rule requires knowing both the priors (of different possible states or hypotheses) and the conditional distribution of feature vector for each possible state
- In some cases, we are given some data/samples and need to design/train a classifier with some general knowledge about the conditional distribution of feature vectors (but **without** knowing the exact distribution)
- Approach: Use the data/samples to "estimate" the unknown probabilities and probability densities and then use the estimated probabilities and probability densities (in place of true values)

Parameter Estimation

- While estimating prior probabilities is not very difficult, estimating conditional densities of feature vectors, especially when the feature vector dimension is large, requires a large set of data without imposing any assumptions

- Approach: Assume that probability densities belong to a family of distributions, which are parametrized by a (small) set of parameters
  - Problem can be cast as one of estimating the parameters of probability densities/distributions
  - Example: Gaussian (or normal) densities

$$p(\mathbf{x}|w_i) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma_i|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right), \quad i = 1, 2, \ldots, c$$

    - Need to estimate the parameters $(\boldsymbol{\mu}_i, \Sigma_i)$

Parameter Estimation

- Two possible scenarios
  - Scenario 1: The parameters can be modeled as random variables with “known” prior distribution
    - Bayesian estimation
    - Samples used to compute the posterior density
  - Scenario 2: Little prior information available regarding the parameters
    - Parameters assumed fixed but unknown
    - Try to find the parameters that are most likely to produce the samples

➡ Maximum likelihood estimates

Maximum Likelihood Estimation

- Suppose that the collection of samples is partitioned according to class (or possible state): $\mathcal{D}_1, \ldots, \mathcal{D}_c$
  - $\mathcal{D}_j$ - samples from class j
  - Samples in $\mathcal{D}_j$ are independent and identically distributed (i.i.d.) with probability density function $p(\cdot|w_j)$
- Key assumption: probability density function $p(\cdot|w_j)$ belongs to a family of probability distributions parametrized by a parameter vector $\boldsymbol{\theta}_j$
  - Example: Gaussian distribution $p(\mathbf{x}|w_j) \sim N(\boldsymbol{\mu}_j, \Sigma_j)$ with $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j, \Sigma_j)$
  - Parameter vector is unknown
  - Typically assumed that the samples from a different class provides no information about the unknown parameters $\boldsymbol{\theta}_j$

Maximum Likelihood Estimation

- Goal: Estimate the unknown parameters $\boldsymbol{\theta}_j$
- Oftentimes, little is known about the unknown parameters and they need to be estimated from available data $\mathcal{D}_j = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ (samples for class w_j)
- <u>Approach</u>: In the absence of additional information, try to find the probability density that maximizes the likelihood

$$p(\mathcal{D}_j|\boldsymbol{\theta}) = \prod_{k=1}^{n} p(\mathbf{x}_k|\boldsymbol{\theta})$$

- Log-likelihood:

$$\boxed{\ell(\boldsymbol{\theta}) \equiv \log\left(p(\mathcal{D}_j|\boldsymbol{\theta})\right) = \sum_{k=1}^{n} \log\left(p(\mathbf{x}_k|\boldsymbol{\theta})\right)}$$

- sum of log-likelihood of samples

Maximum Likelihood Estimation

- Maximum likelihood estimate

$$\boldsymbol{\theta}_{ML} \in \arg\max_{\boldsymbol{\theta}} \; \ell(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{n} \log\left(p(\mathbf{x}_k|\boldsymbol{\theta})\right)$$

  - Parameter values that are “most likely” to produce the available samples

Gaussian Cases

- Suppose $p(\mathbf{x}|w_j) \sim N(\boldsymbol{\mu}, \Sigma)$

$$\log\left(p(\mathbf{x}_k|w_j)\right) = -\frac{1}{2}\log\left((2\pi)^d|\Sigma|\right) - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})$$

- Log-likelihood:

$$\ell(\boldsymbol{\theta}) = -\sum_{k=1}^{n}\left(\frac{1}{2}\log\left((2\pi)^d|\Sigma|\right) + \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})\right)$$

$$\boldsymbol{\theta}_{ML} \in \arg\max_{\boldsymbol{\theta}}\ \ell(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}}\ -\sum_{k=1}^{n}\left(\frac{1}{2}\log\left((2\pi)^d|\Sigma|\right) + \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu})\right)$$

$$\boldsymbol{\theta}_{ML} \in \arg\max_{\boldsymbol{\theta}} \; -\sum_{k=1}^{n} \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_k - \boldsymbol{\mu})$$

Gaussian Cases

- Case 1 - Unknown mean: $\boldsymbol{\theta} = \boldsymbol{\mu}$
  - Gradient of the log-likelihood

$$\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \sum_{k=1}^{n} \Sigma^{-1}(\mathbf{x}_k - \boldsymbol{\mu}) = \Sigma^{-1} \sum_{k=1}^{n} (\mathbf{x}_k - \boldsymbol{\mu})$$

  - Set the gradient to zero (vector) to find the maximizer

$$\boxed{\boldsymbol{\theta}_{ML} = \boldsymbol{\mu}_{ML} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k}$$

Sample Mean !!!

Gaussian Cases

- Case 2 - Unknown mean and covariance matrix: $\boldsymbol{\theta} = (\boldsymbol{\mu}, \Sigma)$

$$\boxed{\boldsymbol{\mu}_{ML} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k, \;\; \Sigma_{ML} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \boldsymbol{\mu}_{ML})(\mathbf{x}_k - \boldsymbol{\mu}_{ML})^T}$$

- Special case: univariate Gaussian $\boldsymbol{\theta} = (\mu, \sigma^2)$

$$\ell(\boldsymbol{\theta}) = -\sum_{k=1}^{n}\left(\frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}(x_k - \mu)^2\right)$$

$$\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}) = \begin{bmatrix} \sum_{k=1}^{n}\frac{x_k-\mu}{\sigma^2} \\ \sum_{k=1}^{n}\left(-\frac{1}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}(x_k - \mu)^2\right) \end{bmatrix} = \mathbf{0} \quad \Rightarrow$$

$$\mu_{ML} = \frac{1}{n}\sum_{k=1}^{n}x_k$$

$$\sigma^2_{ML} = \frac{1}{n}\sum_{k=1}^{n}(x_k - \mu_{ML})^2$$

SCIENCE ACADEMY

DIMENSIONALITY REDUCTION AND CLASSIFICATION: PCA AND DA

DATA/MSML 603: Principles of Machine Learning

Principal Component Analysis

- Suppose that we have feature vectors with *d* features

$$\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$$

- First, assume that we want to approximate them using a single vector $\mathbf{x}_0$

- How should we choose this vector so that we can minimize

$$J_0(\mathbf{x}_0) = \sum_{k=1}^{n} \|\mathbf{x}_0 - \mathbf{x}^k\|^2$$

  - Called "squared error"

Principal Component Analysis

- Answer: Sample mean

$$\mathbf{m} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}^k$$

- Proof:

$$J_0(\mathbf{x}_0) = \sum_{k=1}^{n}\|\mathbf{x}_0 - \mathbf{x}^k\|^2 = \sum_{k=1}^{n}\|(\mathbf{x}_0 - \mathbf{m}) + (\mathbf{m} - \mathbf{x}^k)\|^2$$

$$= \sum_{k=1}^{n}\|\mathbf{x}_0 - \mathbf{m}\|^2 + 2(\mathbf{x}_0 - \mathbf{m})^T\sum_{k=1}^{n}(\mathbf{m} - \mathbf{x}^k) + \sum_{k=1}^{n}\|\mathbf{m} - \mathbf{x}^k\|^2$$

$$= \sum_{k=1}^{n}\|\mathbf{x}_0 - \mathbf{m}\|^2 + \boxed{\sum_{k=1}^{n}\|\mathbf{m} - \mathbf{x}^k\|^2}$$

Independent of $\mathbf{x}_0$

Principal Component Analysis

- Sample mean does not tell us how the samples are distributed around the sample mean, i.e., variability in the data

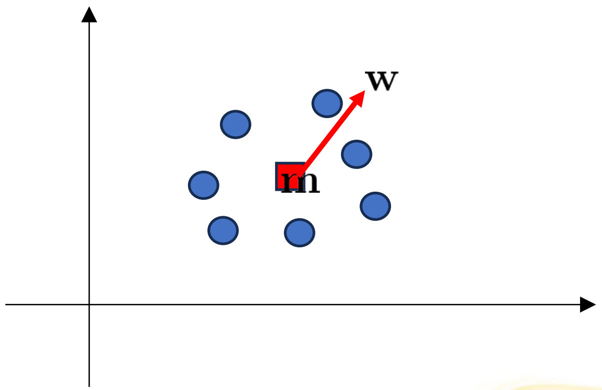
- Let's be a little bit more generous to ourselves

- Suppose that we want to approximate the data using

$$\mathbf{x}_1^k = \mathbf{m} + a_k \mathbf{w}, \quad k = 1, \ldots, n$$

Approximates variation around sample mean

where $\mathbf{w}$ is some vector in $\mathbb{R}^d$ with $\|\mathbf{w}\| = 1$



- We still want to minimize the following squared error

$$J_1(\mathbf{w}, \mathbf{a}) = \sum_{k=1}^{n} \|(\mathbf{m} + a_k \mathbf{w}) - \mathbf{x}^k\|^2, \quad \text{where} \quad \mathbf{a} = (a_k : k = 1, \ldots, n)$$

Principal Component Analysis

Q1: For fixed $\mathbf{w}$, how should we choose a_k so that $\|x^k - (\mathbf{m} + a_k \mathbf{w})\|^2$ is minimized?

Q2: How do we choose $\mathbf{w}$ so that we can minimize the squared error?

$$J_1(\mathbf{w}, \mathbf{a}(\mathbf{w})) = \sum_{k=1}^{n} \|(\mathbf{m} + a_k(\mathbf{w})\mathbf{w}) - \mathbf{x}^k\|^2, \quad \text{where} \;\; \mathbf{a}(\mathbf{w}) = (a_k(\mathbf{w}) : k = 1, \ldots, n) \tag{4-1}$$

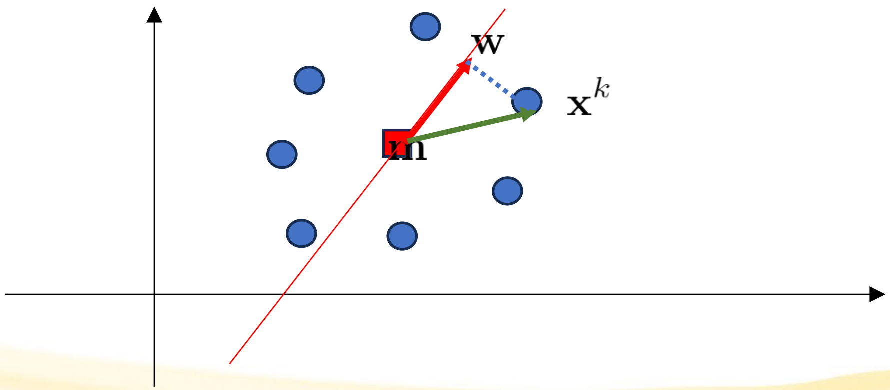
Principal Component Analysis

- Answer to Q1:

$$\begin{aligned}\|\mathbf{m} + a_k\mathbf{w} - \mathbf{x}^k\|^2 &= \|a_k\mathbf{w} - (\mathbf{x}^k - \mathbf{m})\|^2 \\ &= a_k^2\|\mathbf{w}\|^2 - 2a_k\mathbf{w}^T(\mathbf{x}^k - \mathbf{m}) + \|\mathbf{x}^k - \mathbf{m}\|^2 \\ &= a_k^2 - 2a_k\mathbf{w}^T(\mathbf{x}^k - \mathbf{m}) + \|\mathbf{x}^k - \mathbf{m}\|^2\end{aligned} \quad (4\text{-}2)$$

  - Differentiate w.r.t. a_k and set it to zero

$$2a_k - 2\mathbf{w}^T(\mathbf{x}^k - \mathbf{m}) = 0 \Rightarrow \boxed{a_k = \mathbf{w}^T(\mathbf{x}^k - \mathbf{m})} \quad (4\text{-}3)$$



Principal Component Analysis

- Answer to Q2: From (4-1), (4-2) and (4-3)

$$\boxed{a_k = \mathbf{w}^T(\mathbf{x}^k - \mathbf{m})}$$

$$\begin{aligned} J_1(\mathbf{w}) &= \sum_{k=1}^{n}(\mathbf{w}^T(\mathbf{x}^k - \mathbf{m}))^2 - 2\sum_{k=1}^{n}(\mathbf{w}^T(\mathbf{x}^k - \mathbf{m}))^2 + \sum_{k=1}^{n}\|\mathbf{x}^k - \mathbf{m}\|^2 \\ &= -\sum_{k=1}^{n}(\mathbf{w}^T(\mathbf{x}^k - \mathbf{m}))^2 + \sum_{k=1}^{n}\|\mathbf{x}^k - \mathbf{m}\|^2 \\ &= -\sum_{k=1}^{n}\mathbf{w}^T(\mathbf{x}^k - \mathbf{m})(\mathbf{x}^k - \mathbf{m})^T\mathbf{w} + \sum_{k=1}^{n}\|\mathbf{x}^k - \mathbf{m}\|^2 \\ &= -\mathbf{w}^T\mathbf{S}\mathbf{w} + \boxed{\sum_{k=1}^{n}\|\mathbf{x}^k - \mathbf{m}\|^2} \end{aligned}$$

Independent of **w**

where $\mathbf{S} := \sum_{k=1}^{n}(\mathbf{x}^k - \mathbf{m})(\mathbf{x}^k - \mathbf{m})^T$

- called "scatter matrix"

Principal Component Analysis

- We need to maximize $\mathbf{w}^T\mathbf{S}\mathbf{w}$ subject to $\|\mathbf{w}\| = 1$
- Since $\mathbf{S}$ is a positive semidefinite matrix, its eigenvalues are real and non-negative. Furthermore, we can find *d* orthonormal eigenvectors $\mathbf{v}^1, \ldots \mathbf{v}^d$ associated with eigenvalues $\lambda_1, \ldots, \lambda_d$
  - Assume that eigenvalues are arranged by decreasing value
  - Eigenvectors $\mathbf{v}^1, \ldots \mathbf{v}^d$ form an orthonormal basis for $\mathbb{R}^d$

$$\mathbf{w} = \sum_{l=1}^{d} \beta_l \mathbf{v}^l \quad \text{with} \quad \sum_{l=1}^{d} \beta_l^2 = 1 \quad \Rightarrow \quad \mathbf{w}^T\mathbf{S}\mathbf{w} = \left(\sum_{l=1}^{d} \beta_l \mathbf{v}^l\right)^T \left(\sum_{l=1}^{d} \beta_l \lambda_l \mathbf{v}^l\right) = \sum_{l=1}^{d} \lambda_l \beta_l^2$$

- Should choose $\beta_1 = 0, \beta_2 = \cdots = \beta_d = 0$ and $\mathbf{w} = \mathbf{v}^1$ to maximize $\mathbf{w}^T\mathbf{S}\mathbf{w}$

Principal Component Analysis

- What if we ant to approximate the feature vectors using d' vectors

$$\mathbf{x}_{d'}^{k} = \mathbf{m} + \sum_{l=1}^{d'} a_{k,l}\mathbf{w}^{l}$$

- How should we choose $a_{k,l}$ and $\mathbf{w}^{l}$?

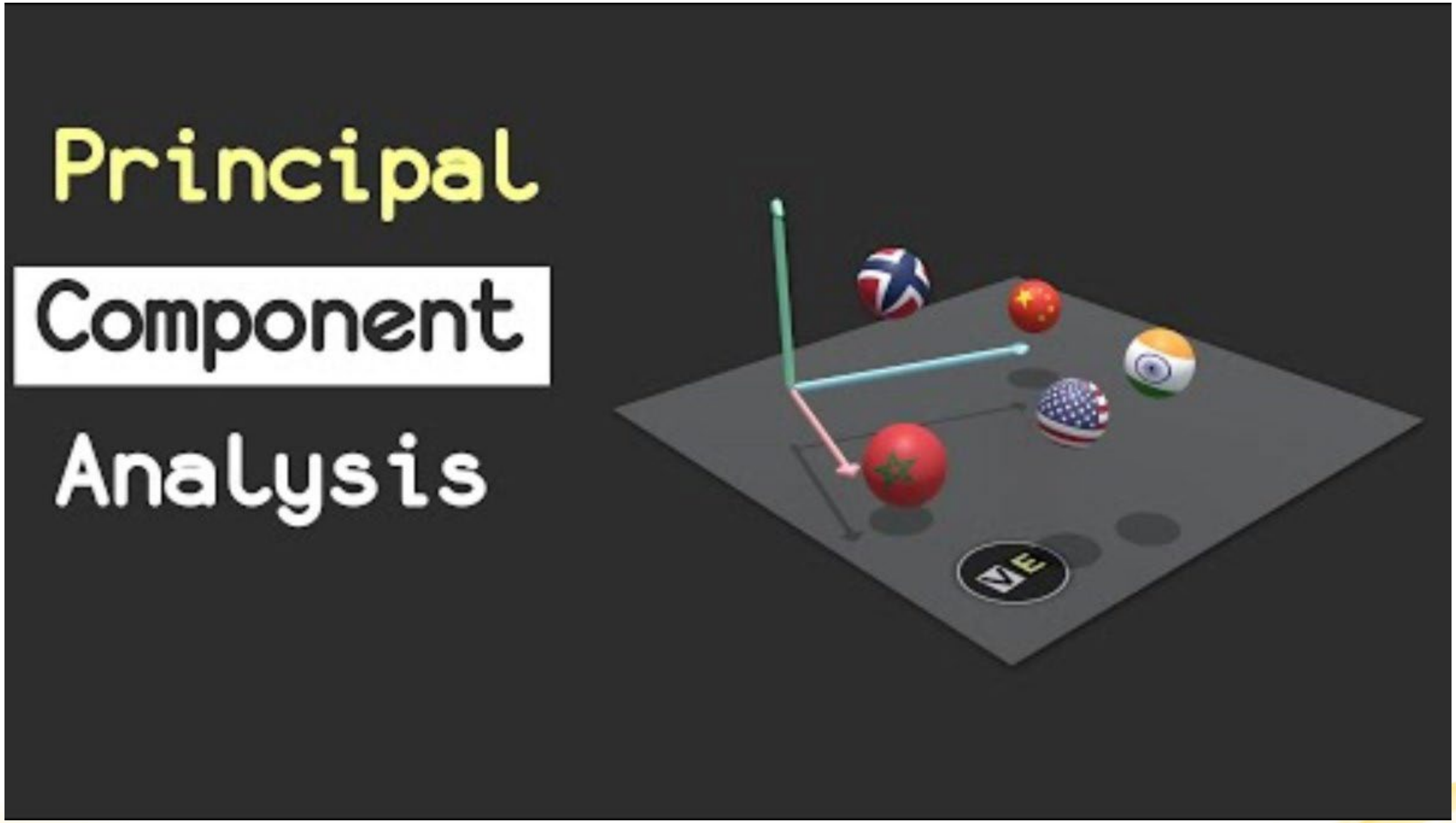
- Answer: We should choose the d' eigenvectors of $\mathbf{S}$ corresponding to the largest eigenvalues and set

$$a_{k,l} = (\mathbf{x}^{k} - \mathbf{m})^{T}\mathbf{w}^{l}$$

Principal Component Analysis

- Why is PCA used in ML?
  - PCA reduces the number of variables or features in a data set while still preserving the most important information like major trends or patterns
  - This reduction can decrease the time needed to train a machine learning model and helps avoid overfitting in a model.

Principal Component Analysis



Principal Component Analysis

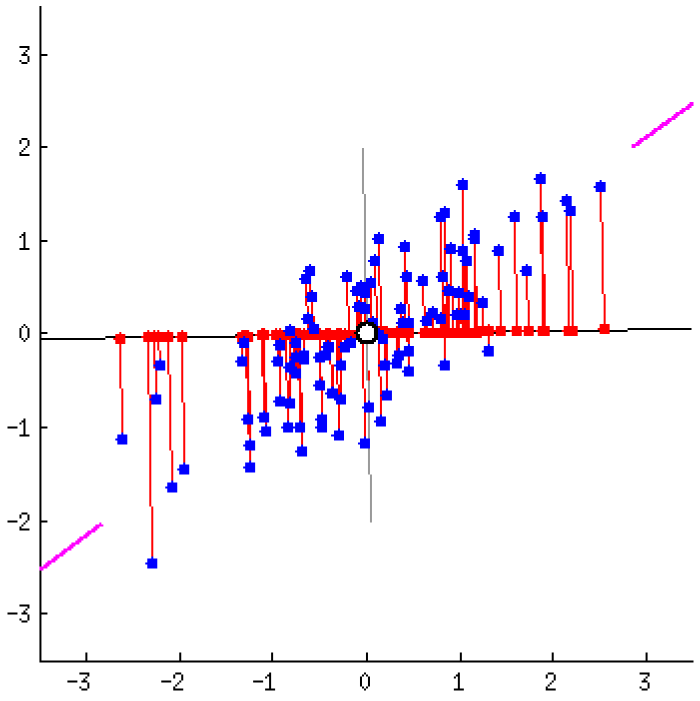
- Matlab examples
  - multinorm_example.m
  - pca_cities.m

Linear Discriminant Analysis

- Dimensionality reduction technique often used for supervised classification
- PCA good for feature vectors of high dimension with new features of lower dimension by taking linear combinations of original features
- But, in general the direction in which there is much variability is not necessarily useful for "discriminating" between samples belonging to different classes (or labels)
- Suppose that we want to "project" data $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n \in \mathbb{R}^d$ (feature vectors) in *d* dimensional feature space onto a line and each feature vector belongs to one of two classes $\mathcal{D}_1$ or $\mathcal{D}_2$
- Question: Which line should we choose?

Linear Discriminant Analysis

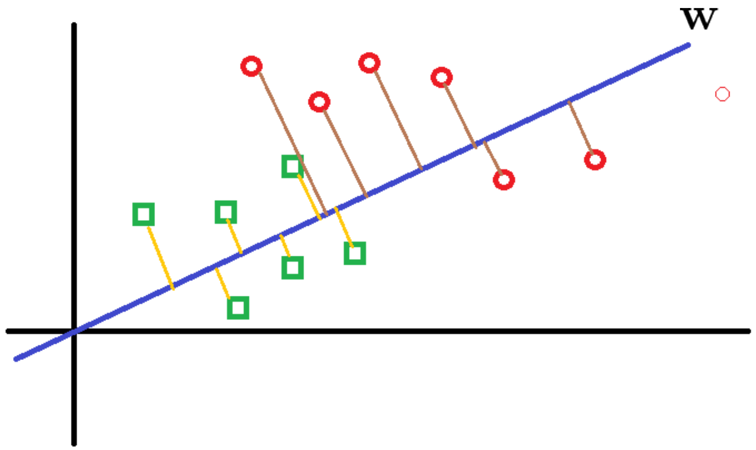
- A picture is worth a thousand words … maybe more …



- Choosing a line is equivalent selecting a unit vector along the line

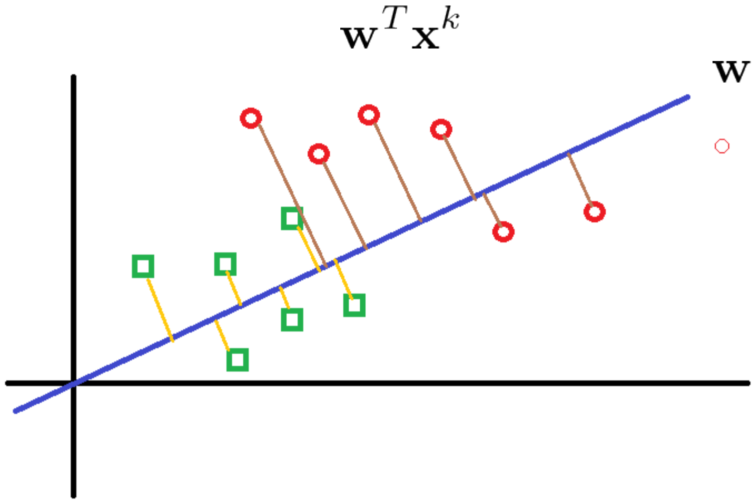
Linear Discriminant Analysis

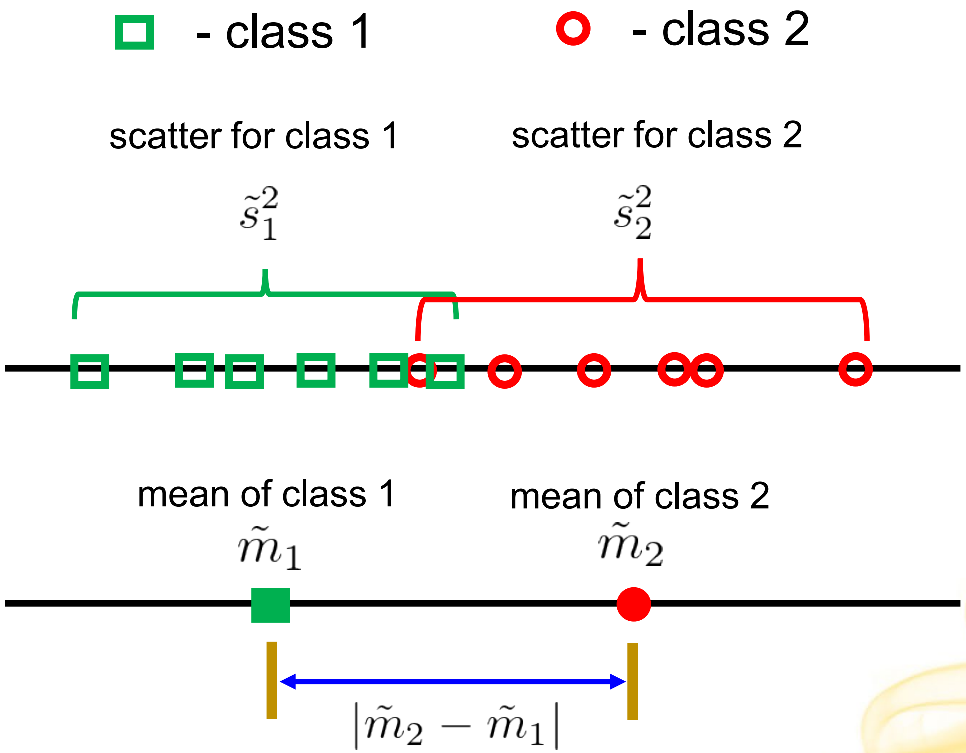
- Assumptions
  - Data/samples have (jointly) Gaussian distributions
  - Data/samples linearly separable, i.e., a straight line or a decision boundary can be drawn to separate data/samples
  - Classes have identical covariance matrix
- Criteria for LDA
  - Maximize the distance between the means of two classes
  - Minimize the variance within individual classes



Linear Discriminant Analysis

- Example





$$\text{maximize } J(\mathbf{w}) = \frac{(\tilde{m}_1 - \tilde{m}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

within-class scatter

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in \mathcal{Y}_i} y, \qquad \tilde{s}_i^2 = \sum_{y \in \mathcal{Y}_i} (y - \tilde{m}_i)^2$$