Quadratic and Linear Discriminant Analysis (QDA and LDA)

we discussed: reduce the size of NN training set by only keeping representative inst.

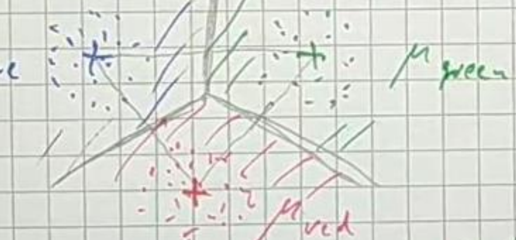
$\Rightarrow$ to the extreme: one representative per class

- basic idea: use the class mean: $k = 1, \dots, C$: N_k: # instances in class k

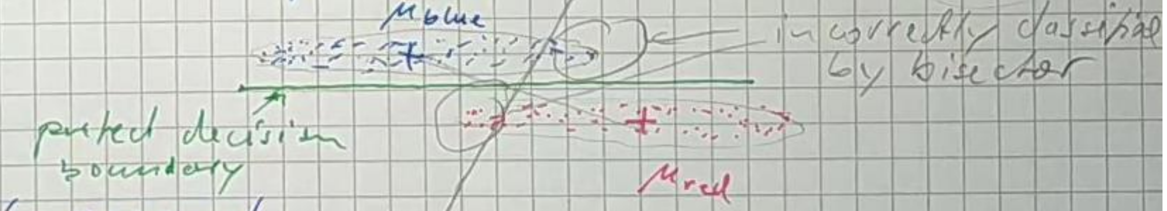
$$\mu_k = \frac{1}{N_k} \sum_{i: Y_i = k} X_i$$

classify query X: $\hat{Y} = \arg\min_k d(X, \mu_k)$

this works, when data form clusters (one cluster per class) and a mean is a good cluster representative

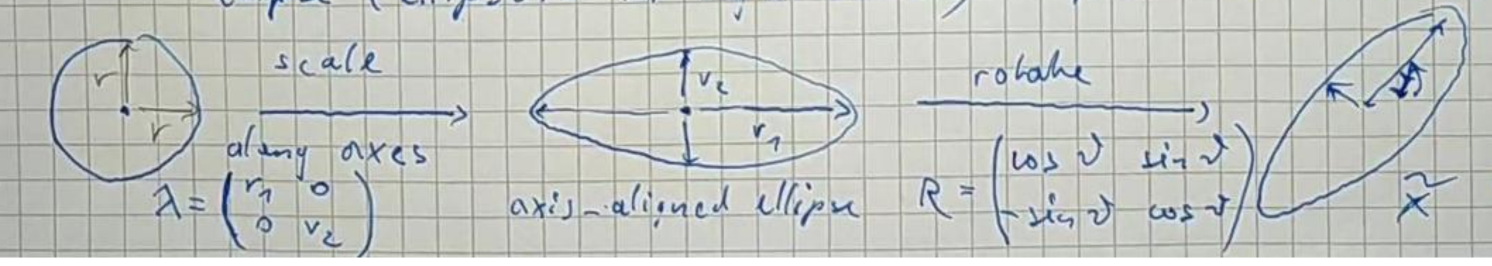


in practice, data are rarely so nice, example where QDA would help:



$\Rightarrow$ need to take cluster shape into account

nearest mean works for circular clusters, next complicated geometric shape ellipse (ellipsoid in higher dims.). from circle to ellipse:



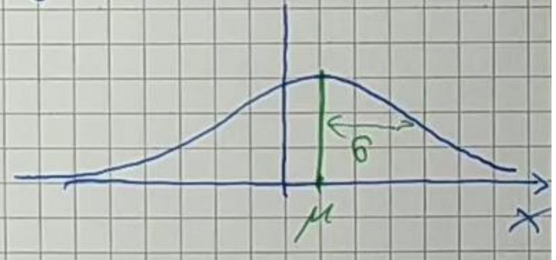
easy to do by matrix multipl.

X $\quad \Lambda = \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix}$ axis-aligned ellipse $\quad R = \begin{pmatrix} \cos\vartheta & \sin\vartheta \\ -\sin\vartheta & \cos\vartheta \end{pmatrix}$ $\quad \tilde{X}$ $\quad \tilde{X} = R \cdot \Lambda \cdot X$

\# of degrees of freedom:
- circle: center (D dimensions) radius (1 dimension)
- ellipse: — " — scaling + rotation $\hat{=}$ symmetric matrix $\frac{D}{2}(D+1)$

$$O(D): (D+1) \quad \text{vs.} \quad O(D^2): D + \frac{D}{2}(D+1) \quad \text{d.o.f.}$$

- How to fit an ellipse to a data cluster
  - consider each class separately $\Rightarrow$ fit C independent ellipses (drop the class label for convenience)
  - central limit theorem of probability theory: the superposition of ~~man~~ infinitely many random events converges to a Gaussian distribution
  - reminder: 1-D Gaussian $\mathcal{N}(\underset{\text{mean}}{\mu}, \underset{\text{variance}}{\sigma^2}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$



  - generalization to arbitrary dimensions:

μ: D-dimensional vector

$\sigma^2 \to \Sigma$: $D \times D$ symmetric positive definite matrix "co-variance matrix"

compute square root: $\Sigma = \Sigma^{1/2}\,\Sigma^{1/2}$ $\qquad \Sigma^{1/2} = \underset{\text{rotation}}{R} \cdot \underset{\text{axis-aligned scaling}}{\lambda}$

$\sigma^2 = \sigma \cdot \sigma$

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{\sqrt{\underset{\text{determinant}}{\det}(2\pi\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

Derivation of QDA (Quadratic Discriminant Analysis)

- intuition: data features are arranged in feature space such that we have an elliptic cluster for each class ⇒ fit a model that captures these clusters
- mathematics: arrangement of the features for a given class is expressed by the likelihood function: $p(X \mid Y=k)$ and prior $\hat{p}(Y=k) = \frac{N_k}{N}$

  ⇒ learn a model for these likelihoods,

  specifically: elliptic clusters $\hat{=}$ multivariate Gaussian

  ⇒ generative classifier ← one cluster per class

$$p(Y=k \mid X) = \frac{p(X \mid Y=k) \cdot p(Y=k)}{p(X)}$$

$p(Y=k \mid X)$ ← generative classifier; $p(Y=k)$ ← how often occurs each class?; $p(X)$ ← normalization

- maximum likelihood principle:
  - we can treat each class in isolation ⇒ simplified $TS = \{X_i\}_{i=1}^{N_k}$ for fixed class k
  - assumption: $TS = \{X_i\}_{i=1}^{N_k}$ is a typical representation of the cluster

    turn assumption around: search for model, such that observed data are "as typical as possible".

    mathematically: $p(TS \mid \text{model}) \to$ maximize, fit the model so

    "maximum likelihood fit" $\hat{=}$ model where the data have max. prob.
  - simplify $p(TS \mid \text{model})$ by the i.i.d. assumption $\hat{=}$ assume that all training instances are independently drawn from the same model $\hat{=}$ factorize prob.

$$\boxed{p(TS \mid \text{model}) = \prod_i p(X_i \mid \text{model})} \Rightarrow \text{maximize via model fit}$$

- specifically when $p(X_i | \underbrace{\text{model}}_{\mu, \Sigma})$ is a multivariate Gaussian

μ, Σ mean and covariance matrix

$$p(x_i | \mu, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \cdot \exp\left(-\frac{1}{2} (X_i - \mu) \Sigma^{-1} (x_i - \mu)^T\right)$$

(X_i is a row vector, therefore μ also row vector $\underbrace{(X_i - \mu) \cdot (X_i - \mu)^T}_{\text{scalar product}}$

- simplify: take the negative logarithm of likelihood

likelihood → max ≙ log likelihood → max ≙ − log likelihood → min

$$-\log \prod_{i=1}^{N} p(X_i | \mu, \Sigma) = \sum_{i=1}^{N} -\log p(x_i | \mu, \Sigma)$$

$$= \sum_{i=1}^{N} -\left[\log \frac{1}{\sqrt{\det(2\pi\Sigma)}} + \left(-\frac{1}{2}(x_i - \mu) \Sigma^{-1} (x_i - \mu)^T\right)\right]$$

$$\text{Loss}(TS | \mu, \Sigma) = \sum_{i=1}^{N} \log \sqrt{\det(2\pi\Sigma)} + \frac{1}{2}(x_i - \mu) \Sigma^{-1} (x_i - \mu)^T \to \min$$

- to find the minimum, take the derivative of the loss w.r.t. parameters and set it to zero ⇒ solve for the parameters ⇒ ~~m~~ estimated params $\hat{\mu}, \hat{\Sigma}$

$$\frac{\partial \text{Loss}}{\partial \mu} = \cancel{\Sigma} - \sum_{i=1}^{N} \Sigma^{-1} (x_i - \mu)^T \overset{!}{=} 0 \quad | -\Sigma \cdot \quad \left[\text{matrix calculus } \frac{\partial v A v^T}{\partial v} = 2 A v^T\right]$$

left multiply

$$\sum_{i=1}^{N} \underbrace{\Sigma \cdot \Sigma^{-1}}_{= I} (x_i - \mu)^T = -\Sigma \cdot 0 = 0$$

$$\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} \mu = \mu \overbrace{\sum_{i=1}^{N} 1}^{= N} = \mu \cdot N$$

$$\Rightarrow \boxed{\mu = \frac{1}{N} \sum_{i=1}^{N} x_i}$$

$\frac{\partial \text{Loss}}{\partial \Sigma}$ is a bit hard, because we actually have Σ^{-1}

$\Rightarrow$ introduce abbreviation $K = \Sigma^{-1}$ precision matrix $S = \sum_{i=1}^{N} z_i^T z_i = \sum_{i=1}^{N} (x_i - \mu)^T (x_i - \mu)$

$z_i = x_i - \mu$

scatter matrix

$$\Rightarrow \frac{\partial \text{Loss}}{\partial K} = \frac{\partial}{\partial K}\left[\sum_{i=1}^{N}\left[\log \sqrt{\underbrace{\det(2\pi K^{-1})}_{=\frac{1}{\det(2\pi K)}}} + \frac{1}{2} z_i K z_i^T\right]\right]$$

$$\underbrace{-\frac{1}{2}\log\det(2\pi K)} = -\frac{1}{2}\log (2\pi)^D \cdot \det(K) = -\frac{1}{2}\log(2\pi)^D - \frac{1}{2}\log\det(K)$$

$$\cancel{= \frac{\partial}{\partial K}\left[-\frac{1}{2}\log(2\pi)^D - \frac{1}{2}\log\det(K) + \frac{1}{2} z_i\right.}$$

$$= \frac{\partial}{\partial K}\sum_{i=1}^{N}\left(-\frac{1}{2}\log(2\pi)^D - \frac{1}{2}\log\det(K) + \frac{1}{2} z_i K z_i^T\right)$$

$$\left[\text{matrix calculus: } \frac{\partial v A v^T}{\partial A} = v^T v \qquad \frac{\partial \log\det A}{\partial A} = (A^T)^{-1}\right]$$

$$\frac{\partial \text{Loss}}{\partial K} = \sum_{i=1}^{N}\left(-\frac{1}{2}(K^T)^{-1} + \frac{1}{2} z_i^T z_i\right) \overset{!}{=} 0 \qquad K^T = K \text{ (K and } \Sigma \text{ are symmetric)}$$

$$\sum_{i=1}^{N} -K^{-1} + \underbrace{\sum_{i=1}^{N} z_i^T z_i}_{S} = 0 \Rightarrow S = N \cdot K^{-1}$$

$$\boxed{\hat{\Sigma} = \frac{1}{N} S}$$

$\frac{1}{N} S$: empirical covariance matrix

$\Rightarrow$ training: for each class k, compute $\hat{\mu}_k = \frac{1}{N_k} \sum_{i: y_i = k} x_i$, $\cancel{\Sigma_k = \frac{1}{N_k} \sum_{i: y_i = k} (x_i - \mu_k)^T}$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i: y_i = k} (x_i - \mu_k)^T \cdot (x_i - \mu_k)$$

• prediction: for a new instance x, compute the likelihood of x for each class, weight with the class prior, return the most probable class

$$\hat{Y} = \arg\max_k p(Y = k \mid X) = \arg\max_k \frac{p(X \mid Y = k)\, p(Y = k)}{p(X)}$$

$$= \arg\min_k -\log p(X \mid Y = k) - \log p(Y = k)$$

$$= \arg\min_k \underbrace{\frac{1}{2} \log \det(2\pi \hat{\Sigma}_k)}_{\text{independent of } X} + \frac{1}{2} (X - \hat{\mu}_k) \hat{\Sigma}_k^{-1} (X - \hat{\mu}_k)^T - \underbrace{\log p(Y = k)}_{\text{independent of } X}$$

$$b_k = \frac{1}{2} \log \det(2\pi \hat{\Sigma}_k) - \log p(Y = k)$$

adjust for cluster shape

QDA: $\boxed{\hat{Y} = \arg\min_k \frac{1}{2} \underbrace{(X - \hat{\mu}_k) \hat{\Sigma}_k^{-1} (X - \hat{\mu}_k)^T}_{\text{squared Mahalanobis distance}} + b_k}$

[scatter matrix is an outer product:

$A = v^T v$

row vector

$A_{ij} = v_i \cdot v_j$]

LDA (linear discriminant analysis)

- simplification of QDA when all clusters have the same shape

$$\Sigma_k = \Sigma_{k'} := \Sigma_w \quad \text{"within-class covariance"}$$

$\Rightarrow$ we can get rid of the quadratic terms in QDA, only linear terms remain

~~replace eq.~~ rewrite eq. with precision matrix

$$\arg\min_k \frac{1}{2} \underbrace{(x-\hat{\mu}_k)\hat{K}_w(x-\hat{\mu}_k)^T}_{} + b_k$$

$$\underbrace{x\hat{K}_w x^T}_{\text{independent of } k} - 2\hat{\mu}_k \hat{K}_w x^T + \underbrace{\hat{\mu}_k \hat{K}_w \hat{\mu}_k^T}_{\text{independent of } x}$$

$\Rightarrow$ has no effect on the $\arg\min_k$

$\Rightarrow$ drop

$\Rightarrow$ absorb into $b_k' = b_k + \frac{1}{2}\hat{\mu}_k \hat{K}_w \hat{\mu}_k^T$

$$\arg\min_k -\underbrace{\hat{\mu}_k \hat{K}_w}_{w_k} x^T + b_k' = \boxed{\arg\max_k w_k x^T - b_k' = \hat{Y}} \quad \text{LDA}$$

- interpretation: case $C=2 \Rightarrow \begin{matrix} w_1 \\ b_1' \end{matrix}$ for class 1, $\begin{matrix} w_0 \\ b_0' \end{matrix}$ for class 0

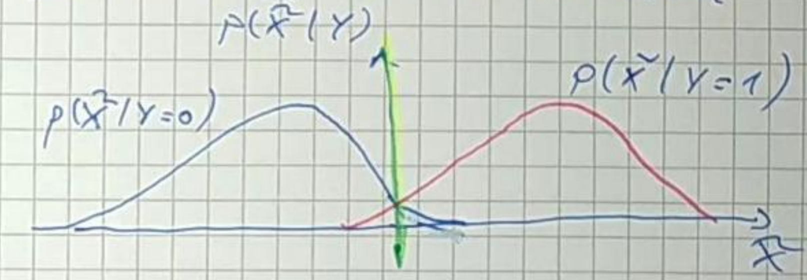
$$\arg\max_{k=0,1} \left(w_1 x^T - b_1',\ w_0 x^T - b_0'\right) \Leftrightarrow (w_1 - w_0)x^T - (b_1' - b_0') = \begin{cases} \geq 0 \Rightarrow \hat{Y} = 1 \\ < 0 \Rightarrow \hat{Y} = 0 \end{cases}$$

simplified rule: LDA, 2 classes $\boxed{\hat{Y} = \text{sign}(w x^T + b)}$ $\quad w = w_1 - w_0 \quad b = b_0' - b_1'$

- in practice, this also works when clusters are not elliptic $\Rightarrow$ choose b by cross-validation an expression of the form $w \cdot X^T + b$ has an intuitive effect:

- projects the data X^T onto the vector w $\Rightarrow$ 1-dimensional feature $w \cdot X^T$
- $+b$ shifts the origin of the new feature such that the decision boundary is at the origin

$$\tilde{X} = \underset{\mathbb{R}^D}{w} \cdot \underset{\mathbb{R}^D}{X^T} + b \in \mathbb{R}$$

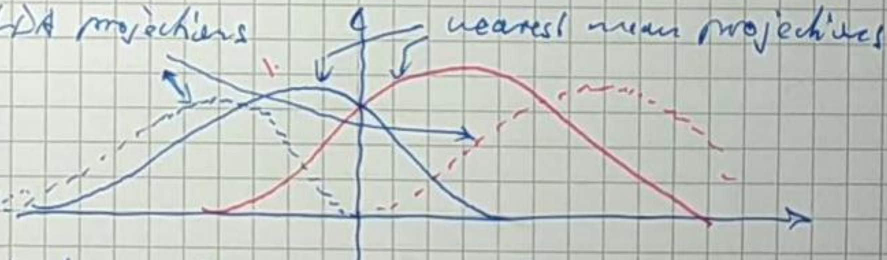


$D = 2$

nearest mean classifier $w = \mu_1 - \mu_0$

$\Rightarrow$ data are projected on the line through the means

$$\boxed{w = (\hat{\mu}_1 - \hat{\mu}_0) \cdot \hat{\Sigma}_w^{-1}}$$



effect: rotates the normal of the decision boundary (relative to the line through the means) such that the cluster shape is considered $\Rightarrow$ less overlaps

numerical example:

$$\mu_1^T = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \quad \mu_0^T = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (\mu_1 - \mu_0)^T = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} (3/2)^2 & 0 \\ 0 & \left(\frac{1}{2}\right)^2 \end{pmatrix} \qquad \Sigma^{-1} = \begin{pmatrix} \frac{4}{9} & 0 \\ 0 & 4 \end{pmatrix}$$

$$w = (2 \;\; -1) \begin{pmatrix} \frac{4}{9} & 0 \\ 0 & 4 \end{pmatrix} = \left(\frac{8}{9} \;\; -4 \right)$$

LDA training alg:

- compute mean of each class: $\hat{\mu}_k = \frac{1}{N_k} \sum_{i: Y_i = k} X_i$

- compute within-class covariance:

$$\Sigma_W = \frac{1}{N} \sum_{i=1}^{N} (X_i - \mu_{Y_i})^T (X_i - \mu_{Y_i})$$

always subtract the mean of the appropriate class

- in 2-class case: $\hat{w} = (\hat{\mu}_1 - \hat{\mu}_0) \cdot \Sigma_W^{-1}$

determine offset $\hat{b}$ such that the training error is minimized for $\tilde{X} = w X^T + b$