Classification

Each unit in a pop is represented by: $(x', L)'$ (L → label $\in \{1, \dots, g\}$; x → vector of features $x = (x_1, \dots, x_p) \in \mathcal{X}$ (e.g. $\mathbb{R}^p$))

Definition (classification)

Learning $\delta: \mathcal{X} \longrightarrow \{1, 2, \dots, g\}$

$\delta(x) = 3 \iff$ the unit for which x is observed belongs to group 3.

Supervised learning (discriminant analysis)

training data:

	x_1	...	x_p	L	
1	x_{11}	...	x_{1p}	e_1	
2	x_{21}		x_{2p}		
:					
sample n	x_n		x_{np}	e_n	$\Longrightarrow$ learn δ

Goal:

Unsupervised learning (cluster analysis)

	x_1	...	x_p	L
1	x_{11}	...	x_{1p}	e_1
2	x_{21}		x_{2p}	
:				
n	x_n		x_{np}	e_n

Goals:

1. Estimate $\hat{e}_1, \dots, \hat{e}_n$ & $\hat{g}$
2. | | | | | |
|---|---|---|---|---|
| 1 | x_{11} | ... | x_{1p} | $\hat{e}_1$ |
| 2 | x_{21} | | x_{2p} | |
| : | | | | |
| n | x_n | | x_{np} | $\hat{e}_n$ |

$\Longrightarrow$ learn δ

Supervised learning

3 basic ingredients

1. Distribution of features: $x \mid L = i \sim f_i(x)$ $\quad i = 1, \dots, g$ $\quad$ density of x in group i
2. Prior distribution: $\mathbb{P}[L = i] = p_i$ $\quad i = 1, \dots, g$ $\quad p_i \geq 0$, $\sum_i p_i = 1$ $\quad$ (context depend.)
3. Costs of misclassification: $c(i|j)$ $\quad i, j = 1, \dots, g$

   → Cost of attributing to group i a unit ω, belonging to group j

   a. $c(i|i) = 0 \quad i = 1, \dots, g$

   b. $c(i|j) \geq 0 \quad i \neq j$

CONTEXT DEPENDENT

Optimal classifier

Obs: $\delta: \mathcal{X} \longrightarrow \{1, \dots, g\}$ is equivalent to a partition $\{R_1, \dots, R_g\}$ of $\mathcal{X}$.

Partition $R_i \subseteq \mathcal{X}$ $i = 1, \dots, g$ s.t.:
- $R_i \cap R_j = \emptyset$ if $i \neq j$
- $\bigcup_{i=1}^{g} R_i = \mathcal{X}$

$R_i = \delta^{-1}(i) = \{x \in \mathcal{X} : \delta(x) = i\}$

Optimality criterion: minimize Expected Cost of Misclassification (ECM)

Example

$g = 2$ (discriminant classifier)

$\delta \longleftrightarrow \{R_1, R_2\}$

$\delta(x) = 1 \iff x \in R_1$

$\delta(x) = 2$ otherwise (indeed $R_2 = R_1^c$)

$\delta: \mathcal{X} \longrightarrow \{1, 2\}$

$$ECM(\delta) = \int_{R_2} c(2|1) f_1(x) p_1 dx + \int_{R_1} c(1|2) f_2(x) p_2 dx =$$

$$= \int_{\mathcal{X}} c(2|1) f_1(x) p_1 dx - \int_{R_1} c(2|1) f_1(x) p_1 dx + \int_{R_1} c(1|2) f_2(x) p_2 dx =$$

$$= c(2|1) p_1 - \int_{R_1} (c(2|1) f_1(x) p_1 - c(1|2) f_2(x) p_2) dx =$$

Optimal δ: $R_1 = \{x \in \mathcal{X} : c(1|2) f_2(x) p_2 \leq c(2|1) f_1(x) p_1\}$

$R_2 = \{x \in \mathcal{X} : c(1|2) f_2(x) p_2 > c(2|1) f_1(x) p_1\}$

General case

$\delta: \mathcal{X} \longrightarrow \{1, 2, \dots, g\}$

$\delta \longleftrightarrow \{R_1, \dots, R_g\}$ part of $\mathcal{X}$

$$\delta(x) = \begin{cases} 1 & x \in R_1 \\ 2 & x \in R_2 \\ \vdots & \\ g & x \in R_g \end{cases}$$

$$ECM(\delta) = \sum_{k\neq 1}\int_{R_k} c(k|1) f_1(x) p_1\,dx + \sum_{k\neq 2}\int_{R_k} c(k|2) f_2(x) p_2\,dx + \dots + \sum_{k\neq g}\int_{R_k} c(k|g) f_g(x) p_g\,dx =$$

$$= \int_{R_1} \sum_{k\neq 1} c(1|k) f_k(x) p_k\,dx + \int_{R_2} \sum_{k\neq 2} c(2|k) f_k(x) p_k\,dx + \dots + \int_{R_g} \sum_{k\neq g} c(g|k) f_k(x) p_k\,dx =$$

OPTIMAL δ:

$$R_1 = \left\{ x\in\chi : \sum_{k\neq 1} c(1|k) f_k(x) p_k \le \sum_{k\neq j} c(j|k) f_k(x) p_k \quad \text{for } j\neq 1 \right\}$$
$$R_2 = \left\{ x\in\chi : \sum_{k\neq 2} c(2|k) f_k(x) p_k \le \sum_{k\neq j} c(j|k) f_k(x) p_k \quad \text{for } j\neq 2 \right\}$$

$$\boxed{\delta(x) = i \iff x\in R_i = \left\{ x\in\chi : \sum_{k\neq i} c(i|k) f_k(x) p_k \le \sum_{k\neq j} c(j|k) f_k(x) p_k \quad \text{for } j\neq i \right\}}$$

Obs: $\delta(x) = t \in \{1,2,\dots,g\} \iff \dfrac{\sum_{k\neq t} c(t|k) f_k(x) p_k}{\sum_j f_j(x) p_j} \le \dfrac{\sum_{k\neq j} c(j|k) f_k(x) p_k}{\sum_j f_j(x) p_j}$

$$\frac{f_k(x) p_k}{\sum_{i=1}^{g} f_i(x) p_i} = \frac{P[X=x \mid L=k]\cdot P[L=k]}{\sum_{i=1}^{g} P[X=x \mid L=i] P[L=i]} = \frac{P[X=x, L=k]}{P[X=x]} = \underbrace{P[L=k \mid X=x]}_{\text{posterior}} \quad \text{(Bayes thm)}$$

$$\boxed{\delta(x) = t \iff \sum_{k\neq t} c(t|k)\, P[L=k \mid X=x] \le \sum_{k\neq j} c(j|k)\, P[L=k \mid X=x]} \quad \text{optimal class.}$$

Special cases

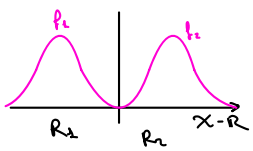
1. $c(i|j) = \text{const} = d > 0$ for $i\neq j$
   $c(i|i) = 0 \qquad i = 1,\dots,g$
   Optimal classifier: $\delta(x) = t \iff$

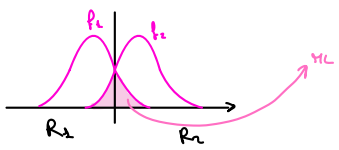
$$\sum_{k\neq t} d\cdot P[L=k \mid X=x] \le \sum_{k\neq j} d\, P[L=k \mid X=x] \qquad j\neq t$$
$$1 - P[L=t \mid X=x] \le 1 - P[L=j \mid X=x]$$
$$\boxed{P[L=t \mid X=x] \ge P[L=s \mid X=x]} \quad \text{BAYES CLASSIFIER}$$

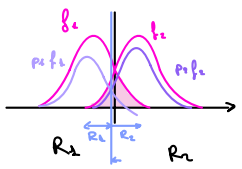
2. $c(i|j) = d > 0 \qquad i\neq j \qquad p_1 = p_2 = \dots = p_g = \frac{1}{g}$
   $c(i|i) = 0 \qquad i = 1,\dots,g$

$$\delta(x) = t \iff \frac{f_t(x) p_t}{\sum f_k(x) p_k} \ge \frac{f_j(x) p_j}{\sum f_k(x) p_k} \iff \boxed{f_t(x) \ge f_j(x) \quad j\neq t} \quad \text{ML CLASSIFIER}$$

Example







Obs: Bayes cl is more flexible than what you might think!

Assume $c(i|j) = c_i \qquad i\neq j$
$c(i|i) = 0 \qquad i = 1,\dots,g$

Classifier: $\delta(x) = t \iff \sum_{k\neq t} c(t|k) f_k(x) p_k \le \sum_{k\neq j} c(j|k) f_k(x) p_k$

$$\sum_{k\neq t} c_k f_k(x) p_k \le \sum_{k\neq j} c_k f_k(x) p_k \longrightarrow \pi_k = \frac{c_k p_k}{\sum c_i p_i} \ge 0\,,\ \sum \pi_k = 1$$
$$\sum_{k\neq t} f_k(x)\pi_k \le \sum_{k\neq j} f_k(x)\pi_k$$

EXERCISE

Work out the Bayes cl. when:
1. $c(i|j) = c_i$ for $i\neq j$
2. $c(i|j) = c_i \cdot h_j \qquad i\neq j \qquad (c^{x_i}\cdot h^{p_j})$

Different costs

$c(t|k) = c_k \ge 0 \qquad t,k = 1,\dots,g$
$c(t|t) = 0$
Opt. Class $\Longrightarrow$ Bayes cl with priors $\pi_j = \dfrac{c_j p_j}{\sum c_k p_k} \qquad j = 1,\dots,g$

EX: $c(i,j) = c^{\alpha_i \beta_j} \quad i,j = 1,\dots,g$

Bayes cl when features are Gaussian

$X \mid L = i \sim N_p(\mu_i, \Sigma_i)$

Bayes cl: $\delta(x) = t \iff \mathbb{P}[L=t \mid X=x] \geq \mathbb{P}[L=s \mid X=x] \quad s = 1,\dots,g$

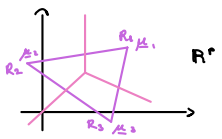
$$f_t(x)\, p_t \geq f_s(x)\, p_s$$

$$\frac{p_t}{\sqrt{(2\pi)^p |\Sigma_t|}} \exp\left[-\tfrac{1}{2}(x-\mu_t)' \Sigma_t^{-1} (x-\mu_t)\right] \geq \frac{p_s}{\sqrt{(2\pi)^p |\Sigma_s|}} \exp\left[-\tfrac{1}{2}(x-\mu_s)' \Sigma_s^{-1} (x-\mu_s)\right]$$

$$\log p_t - \tfrac{1}{2}\log|\Sigma_t| - \underbrace{\tfrac{1}{2}(x-\mu_t)' \Sigma_t^{-1}(x-\mu_t)}_{d^2_{\Sigma_t^{-1}}(x,\mu_t)} \geq \log p_s - \tfrac{1}{2}\log|\Sigma_s| - \underbrace{\tfrac{1}{2}(x-\mu_s)' \Sigma_s^{-1}(x-\mu_s)}_{d^2_{\Sigma_s^{-1}}(x,\mu_s)} \qquad s = 1,\dots,g$$

Obs: If $p_1 = p_2 = \dots = p_g = \frac{1}{g}$ & $\Sigma_1 = \Sigma_2 = \dots = \Sigma_g$

Bayes cl $\implies x \in R_t \iff \delta(x) = t \iff d_{\Sigma^{-1}}(x,\mu_t) \leq d_{\Sigma^{-1}}(x,\mu_s)$

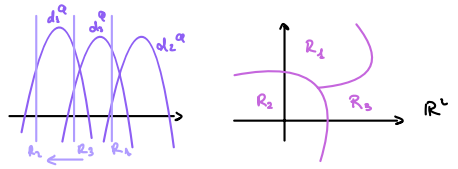


Definition (Quadratic discriminant function)

for $t = 1,\dots,g \quad d_t^Q(x) = \log p_t - \tfrac{1}{2}\log|\Sigma_t| - \tfrac{1}{2}(x-\mu_t)' \Sigma_t^{-1}(x-\mu_t)$

Bayes cl: $\delta(x) = t \iff x \in R_t = \{x \in \mathbb{R}^p : d_t^Q(x) \geq d_s^Q(x)\ \ s = 1,\dots,g\}$

QDA (quadratic disc analysis)



What if: $\Sigma_1 = \Sigma_2 = \dots = \Sigma_g = \Sigma$

$$\delta(x) = t, \quad t = 1,\dots,g \iff \log p_t - \tfrac{1}{2}\log|\Sigma| - \tfrac{1}{2}(x-\mu_t)' \Sigma^{-1}(x-\mu_t) \geq \log p_s - \tfrac{1}{2}\log|\Sigma| - \tfrac{1}{2}(x-\mu_s)' \Sigma^{-1}(x-\mu_s)$$

$$\log p_t - \tfrac{1}{2} x'\Sigma^{-1}x + x'\Sigma^{-1}\mu_t - \tfrac{1}{2}\mu_t'\Sigma^{-1}\mu_t \geq \log p_s - \tfrac{1}{2} x'\Sigma^{-1}x + x'\Sigma^{-1}\mu_s - \tfrac{1}{2}\mu_s'\Sigma^{-1}\mu_s$$

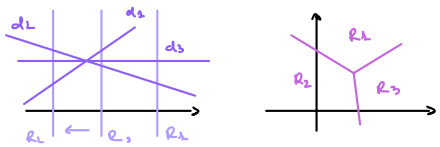
$$x'\Sigma^{-1}\mu_t + \log p_t - \tfrac{1}{2}\mu_t'\Sigma^{-1}\mu_t \geq x'\Sigma^{-1}\mu_s + \log p_s - \tfrac{1}{2}\mu_s'\Sigma^{-1}\mu_s \qquad s = 1,\dots,g$$

Definition (Linear discriminant function)

$t = 1,\dots,g \quad d_t(x) = x'\Sigma^{-1}\mu_t + \log p_t - \tfrac{1}{2}\mu_t'\Sigma^{-1}\mu_t$

Bayes cl: $\delta(x) = t \iff x \in R_t = \{x \in \mathbb{R}^p : d_t(x) \geq d_s(x),\ s = 1,\dots,g\}$

LDA (Lin disc analysis)



Use the training data to estimate f_i $i = 1,\dots,g$, i.e. the distrib. of $X \mid L = i$ $i = 1,\dots,g$

Data: $\begin{matrix} 1 \\ 2 \\ \vdots \\ \vdots \\ n \end{matrix}\begin{bmatrix} x_1' & e_1 \\ x_2' & e_2 \\ \cdot & \cdot \\ \cdot & \cdot \\ x_n' & e_n \end{bmatrix}$

EXAMPLE

LDA: $\mu_1, \dots, \mu_g$, Σ to be estimated from data

QDA: $\mu_1, \dots, \mu_g$, $\Sigma_1, \dots, \Sigma_g$ " " "

Estimate: μ_i with $\bar{x}_i = \frac{1}{n_i} \sum_{j : e_j = i} x_j \qquad n_i = \#\{j : e_j = i\} \qquad i = 1,\dots,g$

In QDA: Σ_i with $S_i = \frac{1}{n_i - 1} \sum_{j : e_j = i} (x_j - \bar{x}_i)(x_j - \bar{x}_i)'$

In LDA: Σ with $S_{pooled} = \frac{1}{n-g} \sum (n_i - 1) S_i$

To estimate $\Sigma_1, \dots, \Sigma_g$ you need $n_1, \dots, n_g$ to be large w.r.t. p.

Class that do not require large sample sizes (parametrize $\Sigma_1, \dots \Sigma_g$):

- **Naive Bayes**

Assume $\Sigma_k = \begin{bmatrix} \sigma_{11}^{(k)} & & 0 \\ & \ddots & \\ 0 & & \sigma_{pp}^{(k)} \end{bmatrix} \implies d_k^Q = \log p_k - \frac{1}{2}\sum_{i=1}^p \log \sigma_{ii}^{(k)} - \frac{1}{2}\sum_{i=1}^p \frac{(x_i - \bar{x}_{ki})^2}{\sqrt{\sigma_{ii}^{(k)}}}$

$\bar{x}_{ki}$ i-th comp. of $\bar{x}_k$

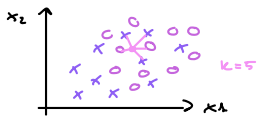
$\sigma_{ii}^{(k)} = \frac{1}{n_k - 1} \sum_{j \in \{1,\dots,n\} : e_j = k} (x_{ji} - \bar{x}_{ki})^2$

KNN Classifier

Fix $K \geq 1$

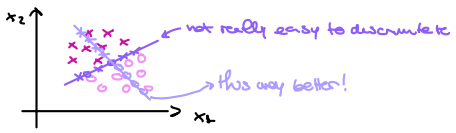
$N_K(x) = \{ k : x_i \text{ in the training set} \}$ ("closest" to x → according to a specified dist)

$\delta(x) = t \iff$ the majority of $x_i \in N_K(x)$ have label t



Fisher's argument for LDA

- Robustness of LDA to Gaussianity ass
- Dimensionality reduction



Assumptions: $i = 1, \dots, g$ $\quad X \mid L = i \sim \mu_i, \Sigma$

Σ same cov in every group i

let $B = \frac{1}{g-1} \sum_{i=1}^{g} (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})'$ $\qquad \bar{\mu} = \frac{1}{g} \sum_{i=1}^{g} \mu_i$

$a \in \mathbb{R}^p$ $\quad X \mid L = i \sim \mu_i, \Sigma$

$\mathbb{E}[a'X \mid L = i] = a'\mu_i$

$\mathrm{Var}(a'X \mid L = i) = a'\Sigma a$

Problem: $\operatorname{argmax}_{a \in \mathbb{R}^p} \frac{a'Ba}{a'\Sigma a} = \operatorname{argmax}_{a \in \mathbb{R}^p} \frac{1}{g-1} \frac{\sum_{i=1}^{g} (a'\mu_i - a'\bar{\mu}_i)^2}{a'\Sigma a}$

$\frac{a'Ba}{a'\Sigma a} = \frac{w'\Sigma^{-1/2} B \Sigma^{-1/2} w}{w'w}$ $\qquad w = \Sigma^{1/2} a$ $\qquad a = \Sigma^{-1/2} w$

$\operatorname{argmax}_{w \in \mathbb{R}^p} \frac{w'\Sigma^{-1/2} B \Sigma^{-1/2} w}{w'w} = e_i$ $\qquad \Sigma^{-1/2} B \Sigma^{-1/2} = \sum_{i=1}^{p} \lambda_i e_i e_i'$

$\operatorname{argmax}_{a \in \mathbb{R}^p} \frac{a'Ba}{a'\Sigma a} = \Sigma^{1/2} e_i$

$a_1 = \Sigma^{-1/2} e_1$

$\vdots$

$a_s = \Sigma^{-1/2} e_s$ $\qquad s = \min(g-1, p)$

$x \to \begin{bmatrix} a_1'x \\ a_2'x \\ \vdots \\ a_s'x \end{bmatrix}$ Fisher discriminant scores

$\mathrm{Cov}(a_k'x, a_j'x) = a_k'\Sigma a_j = e_k'\Sigma^{-1/2}\Sigma\Sigma^{-1/2}e_j = e_j'e_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$

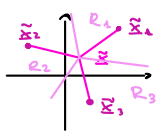
$A = \begin{bmatrix} a_1 \\ \vdots \\ a_s \end{bmatrix}$ $\qquad \mathrm{Cov}(Ax) = I$

How to build a class

μ_i est $\bar{x}_i$

Σ est S_{pooled}

$\bar{x}_i \to \begin{bmatrix} a_1'\bar{x}_i \\ a_2'\bar{x}_i \\ \vdots \\ a_s'\bar{x}_i \end{bmatrix}$ $\quad \bar{x}_i = \sum_{i=1}^{n} x_i$ $\qquad K \leq s$



$\mathbb{R}^K$ $\qquad \delta(x) = ?$ $\qquad x \to \begin{bmatrix} a_1'x \\ \vdots \\ a_s'x \end{bmatrix}$

find the eigenvectors of: $S_{pooled}^{-1/2} \hat{B} S_{pooled}^{-1/2} \longrightarrow \hat{B} = \frac{1}{g-1} \sum_{i=1}^{g} (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})'$

$\delta(x) = t \iff \sum_{i=1}^{K} (\tilde{x}_i - \tilde{x}_{ti})^2 \leq \sum_{i=1}^{K} (\tilde{x}_i - \tilde{x}_{ji})^2$

Evaluating a Classifier

$\mathcal{X} = \begin{bmatrix} x_1 & l_1 \\ \vdots & \vdots \\ x_n & l_n \end{bmatrix}$ training set $\qquad l_i \in \{1, \dots, g\}$ $\quad x_i \in \mathbb{R}^i$ $\quad i = 1, \dots, n$

$\delta : \mathbb{R}^p \longrightarrow \{1, 2, \dots, g\}$ $\quad$ fitted to $\mathcal{X}$

1. Actual error rate (AER)

$\delta \longleftrightarrow \{R_1, \dots, R_g\}$ partition of $\mathbb{R}^p$

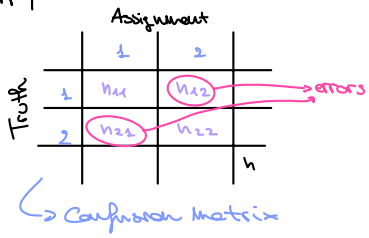
$AER(\delta) = \sum_{k \neq 1} \int_{R_k} f_1(x) p_1 \, dx + \sum_{k \neq 2} \int_{R_k} f_2(x) p_2 \, dx + \dots + \sum_{k \neq g} \int_{R_k} f_g(x) p_g \, dx$

How to estimate AER using data?

wlog let $g = 2$

$AER(\delta) = \int_{R_2} f_1(x) p_1 \, dx + \int_{R_1} f_2(x) p_2 \, dx$

Naive idea: Apply δ to $\mathcal{X}$ and count the mistakes for $i=1,\dots,n$ $\delta(\underline{x}_i)$ and then check if $\delta(\underline{x}_i) = \ell_i$ or not.



↳ Confusion matrix

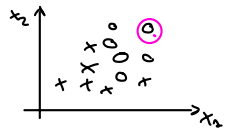
Apparent error rate:

$\widehat{APER}(\delta) = APER(\delta) = \frac{n_{12}+n_{21}}{n}$! $= \overline{err}$ = training error

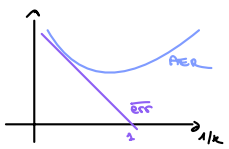
TOO OPTIMISTIC, you tend to overfit the data to make it smaller, BAD!

EXAMPLE

KNN with $K=1$



APER = 0 overfitting the model



That means that we cannot rely on the $\overline{err}$ cause it is NOT the same.

$$\overline{err} = \frac{n_{12}+n_{22}}{n} = \frac{n_{12}}{n} + \frac{n_{22}}{n} = \frac{n_{1\cdot}}{n}\cdot\frac{n_{12}}{n_{1\cdot}} + \frac{n_{21}}{n_{2\cdot}}\cdot\frac{n_{2\cdot}}{n} =$$

$$! = \hat{p}_1 \int_{R_2} f_2(\underline{x})\,d\underline{x} + \hat{p}_2 \int_{R_1} f_2(\underline{x})\,d\underline{x}$$

↳ $\frac{n_{1\cdot}}{n}$ ↳ $\frac{n_{12}}{n_{1\cdot}}$

Obs: Dichotomous class

$\delta(\underline{x}) = \begin{cases} 1 & \text{positive} \\ 0 & \text{negative} \end{cases}$

Confusion matrix:

$APER = \frac{n_{10}+n_{01}}{n}$

Precision: $\frac{n_{11}}{n_{\cdot 1}}$ PPV positive predicted value

Recall: $\frac{n_{11}}{n_{1\cdot}}$ sensitivity

Specificity: $\frac{n_{00}}{n_{\cdot 0}}$

Precision ↑ False Pos ↓

Recall ↑ False Neg ↓

For a more realistic estimate of AER (and the other indexes computed out of the confusion matrix) USE TEST DATA

$$\tilde{\mathcal{X}} = \begin{bmatrix} \underline{\tilde{x}}_1 & \tilde{\ell}_1 \\ \vdots & \vdots \\ \underline{\tilde{x}}_m & \tilde{\ell}_m \end{bmatrix}$$

Apply $\delta_{\mathcal{X}}$ (learned with $\mathcal{X}$)

$$\widehat{AER}(\delta_{\mathcal{X}}) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\left[\delta_{\mathcal{X}}(\underline{\tilde{x}}_i) \neq \tilde{\ell}_i\right]$$

$$\mathbb{1}\left[\delta_{\mathcal{X}}(\underline{\tilde{x}}_i) \neq \tilde{\ell}_i\right] = \begin{cases} 1 & \text{if } \delta_{\mathcal{X}}(\underline{\tilde{x}}_i) \neq \tilde{\ell}_i \\ 0 & \text{if } \delta_{\mathcal{X}}(\underline{\tilde{x}}_i) = \tilde{\ell}_i \end{cases}$$

2. CROSS VALIDATION

Leave One Out

$$\mathcal{X} = \begin{bmatrix} \underline{x}_1' & \ell_1 \\ \vdots & \vdots \\ \underline{x}_n' & \ell_n \end{bmatrix}$$

training data

For $i = 1, \dots, n$

step 1: take unit i out $\quad \mathcal{X}_{-i} = \begin{bmatrix} \underline{x}_1' & \ell_1 \\ \vdots & \vdots \\ \underline{x}_n' & \ell_n \end{bmatrix}$ (row $\underline{x}_i'$, ℓ_i crossed out)

step 2: train δ on $\mathcal{X}_{-i} \Longrightarrow \delta_{-i} : \mathbb{R}^p \longrightarrow \{1, \dots, g\}$

step 3: Evaluate δ_{-i} on unit i $\quad E_i = \mathbb{1}\left[\delta_{-i}(\underline{x}_i) \neq \ell_i\right]$

Return: $\widehat{AER}(\delta_{\mathcal{X}}) = \frac{1}{n}\sum_{i=1}^{n} E_i$ ← low bias - high variability

3. K-fold cross-validation

Fix $K \geq 1$, split the training set in K-subsections (usually 5/10, but it's up to us)

$\mathcal{X} =$ | 1 | / | 2 | / | 3 | / | 4 | $\quad K=4$

By permuting the rows of $\mathcal{X}$ you are randomly assigning units to the K subsets

Obs: try to balance the K subsets

Step 1 : train δ on $\mathcal{X}_{-part\,j} \Longrightarrow \delta_{-part\,j} : \mathbb{R}^p \longrightarrow \{1, \dots, g\}$

Step 2 : Apply $\delta_{-part\,j}$ to part $j \Longrightarrow Err\,j = \frac{1}{n_j} \sum_{i \in part\,j} \varepsilon_i \qquad \varepsilon_i = \mathbb{1}[\delta_{-part\,j}(x_i) \neq e_i] \qquad n_j = \#\{i : i \in part\,j\}$

Return : $\widehat{AER}(\delta_{\mathcal{X}}) = \frac{1}{n} \sum_{j=1}^{k} n_j\, Err\,j$

Obs: Can be used to estimate : Precision, Recall, ...

Obs: leave-one-out $\Longleftrightarrow K = n$

Given K : repeat K-fold cross-val B times, B large. (repeat = permute and then repeat, it changes!)

$\widehat{AER}_1(\delta_{\mathcal{X}}), \dots, \widehat{AER}_B(\delta_{\mathcal{X}}) \longrightarrow \mathbb{E}[\widehat{AER}(\delta_{\mathcal{X}})] = \widehat{AER}_m(\delta_{\mathcal{X}}) = \frac{1}{B} \sum_{i=1}^{B} \widehat{AER}_i(\delta_{\mathcal{X}})$

$\longrightarrow Var[\widehat{AER}(\delta_{\mathcal{X}})] = \frac{1}{B-1} \sum_{i=1}^{B} [\widehat{AER}_i(\delta_{\mathcal{X}}) - \widehat{AER}_m(\delta_{\mathcal{X}})]^2$

$\longrightarrow CI_{1-\alpha}(\mathbb{E}[\widehat{AER}(\delta_{\mathcal{X}})]) = \left[\widehat{AER}_m(\delta_{\mathcal{X}}) \pm \sqrt{\frac{Var[\widehat{AER}(\delta_{\mathcal{X}})]}{B}}\, z_{1-\frac{\alpha}{2}}\right]$

Obs: leave-one-out $\widehat{AER}(\delta_{\mathcal{X}}) = \frac{1}{n} \sum_{i=1}^{n} \varepsilon_i$

$\frac{Var(\varepsilon_i)}{n} \overset{?}{\simeq} Var\left(\frac{1}{n} \sum_{i=1}^{n} \varepsilon_i\right) \overset{?}{<<} Var(\varepsilon_i)$

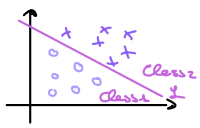
$\delta_{-1}, \delta_{-2}, \dots, \delta_{-n}$ Strongly dependent !! $\Longrightarrow Var\left(\frac{1}{n} \sum \varepsilon_i\right) \simeq Var(\varepsilon_i)$

$\varepsilon_1 \quad \varepsilon_2 \quad \varepsilon_n$

SUPPORT VECTOR MACHINES

Dichotomous Problem

$\mathcal{X} = \begin{bmatrix} x_1' & e_1 \\ \vdots & \\ x_n' & e_n \end{bmatrix} \qquad x_i \in \mathbb{R}^p \quad e_i \in \{1, 2\} \quad i = 1, \dots, n$



Q1: Given $A, B \subseteq \mathbb{R}^p$ when can they be separated by an hyperplane?

Assumptions: Let $CH(A)$ and $CH(B)$ be the convex hull generated by A and B

1. $CH(A) \neq \emptyset$ and $CH(B) \neq \emptyset$
2. $CH(A) \cap CH(B) = \emptyset$
3. Either $CH(A)$ or $CH(B)$ is open

$\overset{Th}{\Longrightarrow} \exists$ separating hyperplan

Obs: Condition 3 can be substituted by:

3'. $CH(A)$ and $CH(B)$ are closed and at least one of them is compact. (satisfied by the training data)

Hyperplanes

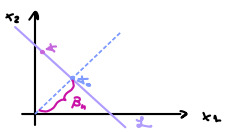
$\mathcal{L}$ hyperplane in $\mathbb{R}^p$

affine subspace of dim $p-1$

Let $\beta \in \mathbb{R}^p$, $\|\beta\| = 1$ s.t. $\beta \perp \mathcal{L}$

$x_0 = span(\beta) \cap \mathcal{L}$

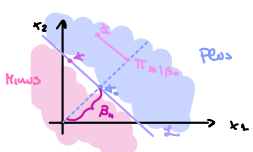
$\beta_0 = \|x_0\|$



$\mathbb{R}^p \ni x \in \mathcal{L} \Longleftrightarrow \pi_{x|\beta} = x_0 \Longrightarrow \frac{\beta\beta'}{\beta'\beta} x = x_0 \Longrightarrow \boxed{\beta' x = \beta_0}$

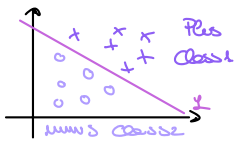
$\forall x \notin \mathcal{L} \quad \beta' x > \beta_0$

$\beta' x < \beta_0$



Obs: $\beta' x - \beta_0$ distance with sign of x from $\mathcal{L}$.

Going back to classification



Assume $\mathcal{X}$ are s.t. $\exists\, \mathcal{L}$ separating Class 1 and Class 2.

$\mathcal{L} \Longrightarrow$ defines R_1 and R_2 half-spaces

$\Longrightarrow$ classifier

let $y_i = +1$ if unit i belongs to class 1.

$y_i = -1$ if unit i belongs to class 2.

Note: $y_i \cdot (\beta' x_i - \beta_0) \geq 0$

distance between x_i and L

$M_L = \min \{ y_i(\beta' x_i - \beta_0) : i = 1, \dots, n \}$

↳ Problem: find L (i.e. β, β_0) s.t. M_L is max

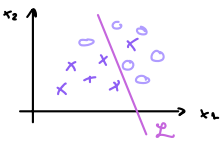
$$\max_{\beta, \beta_0} M \quad \text{s.t.} \quad \|\beta\| = 1 \qquad \text{(OPTIMIZATION PROBLEM)}$$
$$y_i(\beta' x_i - \beta_0) \geq M \quad \text{for} \quad i = 1, \dots, n$$

Solution: $\hat{\beta} = \sum_{i=1}^{n} \hat{\lambda}_i y_i x_i \qquad \hat{\beta}_0 \in \mathbb{R}$

$\hat{f}(x) = \hat{\beta}' x - \hat{\beta}_0 = \sum_{i=1}^{n} \lambda_i y_i x_i' x - \hat{\beta}_0$

Classifier: $C(x) = \text{sgn}(\hat{f}(x))$

What if:



not as easy to introduce an hyper plane

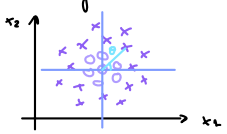
Soft solution: $\max_{\beta, \beta_0} M \quad \text{s.t.} \quad \|\beta\| = 1$

$y_i(\beta' x_i - \beta_0) \geq (1 - \varepsilon_i) M$

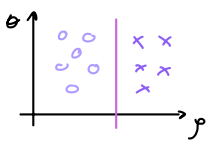
$\varepsilon_i \geq 0, \quad \sum_{i=1}^{n} \varepsilon_i \leq C$ ⟶ Budget cost

Extra parameters:
- C budget
- $C = 0 \implies$ hard problem
- C very large $\implies$ any L would do

What if?



$f = \sqrt{(x_i - \bar{x}_1)^2 + (x_2 - \bar{x}_2)}$



Take $h_1 : \mathbb{R}^p \to \mathbb{R}, \dots, h_m : \mathbb{R}^p \to \mathbb{R}$

and let $h(x) = (h_1(x), \dots, h_m(x))' \in \mathbb{R}^m$

Now transform set:

$$\tilde{X} = \begin{bmatrix} h'(x_1) & y_1 \\ \vdots & \vdots \\ h'(x_n) & y_n \end{bmatrix} \implies \text{find soft or hard separation hyperplane for } \tilde{X}$$

Solution: $\hat{\beta} = \sum_{i=1}^{n} \lambda_i y_i h(x_i)$

$x \in \mathbb{R}^p \qquad \hat{f}(x) = \beta' h(x) - \beta_0 = \sum_{i=1}^{n} \hat{\lambda}_i y_i \underbrace{h'(x_i) h(x)}_{< h'(x_i), h(x) >} - \hat{\beta}_0$

$C(x) = \text{sign}(\hat{f}(x))$

Idea: K (kernel): $\mathbb{R}^p \times \mathbb{R}^p \longrightarrow \mathbb{R}$ s.t. $K(x, w) = \langle h(x), h(w) \rangle = h'(x) \cdot h(w) \quad \forall x, w \in \mathbb{R}^p$

$\implies \hat{f}(x) = \sum_{i=1}^{n} \hat{\lambda}_i y_i K(x_i, x)$ RKHS

Examples of kernels
- $K(x, w) = [1 + x' w]^d$ — d-degree polynomials
- $K(x, w) = \exp[-\gamma \|x - w\|^2]$ — radial basis
- $K(x, w) = \tanh[k_1 x' w + k_2]$ — neural networks