# CLASSIFICATION

21/04

We observe some features of a statistical unit and then we use the informations to attribute the statistical unit to a certain group.

Each statistical unit is represented by $(\underline{X}, L)$ where:

$\underline{X} = [X_1, \dots, X_p] \in \mathcal{X}$ (e.g. $\mathbb{R}^p$) — features

quantitative or qualitative (what we observe for a statistical unit)

$L \in \{1, 2, \dots, g\}$ — labels declaring membership to a group

**Goal:** find $\delta : \mathcal{X} \longrightarrow \{1, 2, \dots, g\}$ — classifier: "you give me the features, I'll tell you the label"

## SUPERVISED

Available a training set, i.e. data in this form:

$$\mathbb{X} = \begin{array}{c} X_1 \; X_2 \; \dots \; X_p \; L \\ \left[\begin{array}{cccc} x_{11} & x_{12} & \dots & x_{1p} \; l_1 \\ x_{21} & x_{22} & \dots & x_{2p} \; l_2 \\ \vdots & & & \\ x_{n1} & x_{n2} & \dots & x_{np} \; l_n \end{array}\right] \end{array} = \begin{array}{c} \underline{X} \; L \\ \left[\begin{array}{cc} \underline{x}_1^T & l_1 \\ \vdots & \\ \underline{x}_n^T & l_n \end{array}\right] \end{array}$$

$\mathbb{X}$: data frame

$\underline{x}_i \in \mathcal{X}$, $l_i \in \{1, 2, \dots, g\}$

$\underline{X}$ observable, $L$ observable

**Goal:** learn the "optimal" $\delta$ based on: 1. model 2. data $\mathbb{X}$ — (DISCRIMINANT ANALYSIS)

## UNSUPERVISED

Available a training set in the form:

$$\mathbb{X} = \begin{array}{c} \underline{X} \; L \\ \left[\begin{array}{cc} \underline{x}_1^T & l_1 \\ \underline{x}_2^T & l_2 \\ \vdots & \\ \underline{x}_n^T & l_n \end{array}\right] \end{array}$$

$\underline{x}_i \in \mathcal{X}$, $l_i \in \{1, \dots, g\}$

$\underline{X}$ observable, $L$ hidden

we believe there are labels but we don't know them (we want to try to group all the observations)

it's good to have a coalization of these two problems (discriminant & clustering analysis) because often happen that we have a training set for which for some unit there's a label, for others no.

**Goal:** a) estimate the labels $\hat{l}_1, \dots, \hat{l}_g$ (and $g$)

(CLUSTER ANALYSIS): $\mathbb{X} \to \hat{\mathbb{X}}$

b) find the optimal $\delta : \mathcal{X} \to \{1, \dots, \hat{g}\}$ based on:

1. model
2. data $\hat{\mathbb{X}} = \begin{bmatrix} \underline{x}_1^T & \hat{l}_1 \\ \vdots & \vdots \\ \underline{x}_n^T & \hat{l}_n \end{bmatrix}$

(DISCRIMINANT ANALYSIS)

**NOTE:** sometimes we use discriminant analysis to see how robust is the classification.

# INGREDIENTS FOR (SUPERVISED) MODEL FOR CLASSIFICATION

We focus just on the model (not on the training set). Then with training set we'll estimate the pieces of the model that we don't know about.

1. $\underline{X} \mid L = i \sim f_i(\underline{x})$ (for simplicity: $X = \mathbb{R}^p \Rightarrow f_i : \mathbb{R}^p \to [0, \infty)$ density)

What is the distribution of the features in any given group?

They must be **different!** If they're not, there's no hope that by looking at the features we'll be able to recognize which table to put on.
How do we check? ANOVA / MANOVA.

2. Prior probabilities

$$\mathbb{P}(L = i) = p_i \qquad i = 1, \dots, g \qquad \text{s.t.} \quad p_i \geq 0, \ \sum_{i=1}^{g} p_i = 1$$

(context dependent) The same problem of classification, the same optimal goal, with the same optimal classifier may have a different solution depending on the BOUNDARY CONDITIONS ("context conditions")

3. Cost of misclassification

| Attributed label (by $\delta$) \ True label | 1 | 2 | 3 | ... | g |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | $c(i\|j)$ | | |
| ⋮ | | | | | |
| g | | | | | |

$c(i|j)$ = cost for attributing to $i$ an unit belonging to group $j$

$c(i|j) \geq 0 \qquad \forall i, j = 1, \dots, g$

$c(i|i) = 0 \qquad \forall i = 1, \dots, g$

(! We're not requiring symmetry ($c(i|j) = c(j|i)$))

If you see a person and you think it's dangerous when it's not, then ok. But if you don't think it's dangerous and it is, then it's a problem. Note that the same problem during the day and during the night is different: the probability of this person of being dangerous will be higher during night (it's subjective!) $\Rightarrow$ what change are the prior probabilities ("boundary/context conditions")

# OPTIMALITY CRITERION FOR CHOOSING $\delta$

Note that: we said that the classifier is a function, it's equivalent to say that it's a partition of the features' space.

Note that:

If $\delta : X \to \{1, \dots, g\}$ we can define: ($\delta$ measurable)

$$R_i = \{ \underline{x} \in X : \delta(\underline{x}) = i \} = \delta^{-1}(i) \qquad \text{(Borel)}$$

$\{R_1, R_2, \dots, R_g\}$ is a partition of $X$:

1. $R_i \cap R_j = \emptyset$ if $i \neq j$
2. $\bigcup_{i=1}^{g} R_i = X$

$\Rightarrow \quad \delta \longleftrightarrow \{R_1, \dots, R_g\}$ (equivalent)

specifying one is equivalent to specify the other

for simplicity assume: $X = \mathbb{R}^p$ and $g = 2$

$\Rightarrow \quad \delta : \mathbb{R}^p \to \{1, 2\}$

$R_1 = \{ \underline{x} \in \mathbb{R}^p : \delta(\underline{x}) = 1 \}$

$R_2 = \{ \underline{x} \in \mathbb{R}^p : \delta(\underline{x}) = 2 \} = R_1^c$

Expected cost of misclassification of $\delta$:

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

with probability $p_1$ an individual belongs to group 1. this individual is showing us some features (some $\underline{x}$) and with probability $f_1(\underline{x}) \, d\underline{x}$ the feature that the individual is showing is exactly the $\underline{x}$ that we wrote. if we misclassify the individual (that belongs to group 1), we pay a cost $c(2|1)$. When do we misclassify it? When our features $\underline{x}$ belongs to $R_2$. (so when $\delta(\underline{x}) = 2$ when $\underline{x}$ is in group 1.)

**Optimization problem:**

find $\delta$ which minimizes $ECM(\delta)$.

($\equiv$ find $R_1, R_2$ s.t. min $ECM(\delta)$)

$$ECM(\delta) = \int_{\mathbb{R}^p} c(2|1) f_1(\underline{x}) p_1 \, d\underline{x} - \int_{R_2} c(2|1) f_1(\underline{x}) p_1 \, d\underline{x} +$$

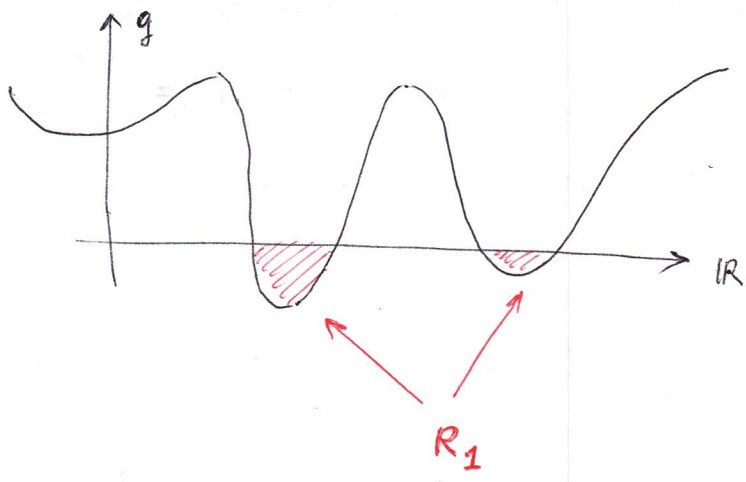
$$+ \int_{R_1} c(1|2) f_2(\underline{x}) p_2 \, d\underline{x}$$

$:= g(\underline{x})$ = cost that we'll pay if we're putting $\underline{x}$ in 1 and we're wrong

$$= c(2|1) p_1 + \int_{R_1} \left( c(1|2) f_2(\underline{x}) p_2 - c(2|1) f_1(\underline{x}) p_1 \right) d\underline{x}$$

how to identify $R_1$ s.t. ECM is min?

$$= c(2|1) p_1 + \int_{R_1} g(\underline{x}) \, d\underline{x} = \text{fixed cost} + \int_{R_1} g(\underline{x}) \, d\underline{x}$$




How to chose $R_1$ (which is the only thing we've free to chose)?

We're always paying something ($c(2|1) p_1$), so we would like to substract something since we want min $ECM(\delta)$.

$\Rightarrow$ optimal $R_1, R_2$:

$$R_1 = \{ \underline{x} \in \mathbb{R}^p : g(\underline{x}) \leq 0 \}$$

$$\leq \{ \underline{x} \in \mathbb{R}^p : c(1|2) f_2(\underline{x}) p_2 \leq c(2|1) f_1(\underline{x}) p_1 \} = \left\{ \underline{x} \in \mathbb{R}^p \text{ s.t. : } \text{cost}\left(\begin{matrix}\text{error: } \underline{x} \text{ is 2,} \\ \text{we put 1}\end{matrix}\right) \leq \text{cost}\left(\begin{matrix}\text{error: } \underline{x} \text{ is 1} \\ \text{we put 2}\end{matrix}\right) \right\}$$

$$R_2 = \{ \underline{x} \in \mathbb{R}^p : c(2|1) f_1(\underline{x}) p_1 \leq c(1|2) f_2(\underline{x}) p_2 \}$$

$\Rightarrow$ optimal $\delta$:

$$\delta(\underline{x}) = \begin{cases} 1 & \text{if } \underline{x} \in R_1 \\ 2 & \text{if } \underline{x} \in R_2 \end{cases}$$

$\longrightarrow$ we're attributing to group 1 those statistical units for which we'll have to pay less if we're making a mistake

if we classify $\underline{x}$ in 1 and it's 2 $\longrightarrow$ cost 1
if we classify $\underline{x}$ in 2 and it's 1 $\longrightarrow$ cost 2
$\underline{x} \in R_1 \iff$ cost 1 < cost 2

General case: $g \geq 2$

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

$$ECM(\delta) = \sum_{k=2}^{g} \int_{R_k} c(k|1) f_1(\underline{x}) p_1 \, d\underline{x} + \sum_{\substack{k=1 \\ k \neq 2}}^{g} \int_{R_k} c(k|2) f_2(\underline{x}) p_2 \, d\underline{x} +$$

$$+ \dots + \boxed{\sum_{k=1}^{g-1} \int_{R_k} c(k|g) f_g(\underline{x}) p_g \, d\underline{x}}$$

for a unit of the group $g$ we can make $g-1$ mistakes, that's why $\sum_{k=1}^{g-1}(\dots)$

$$= \int_{R_1} \underbrace{\sum_{k=2}^{g} c(1|k) f_k(\underline{x}) p_k}_{\text{what we pay if we put } \underline{x} \text{ in 1 and it's not}} d\underline{x} + \int_{R_2} \sum_{\substack{k=1 \\ k \neq 2}}^{g} c(2|k) f_k(\underline{x}) p_k \, d\underline{x} +$$

$$+ \dots + \int_{R_g} \underbrace{\sum_{k=1}^{g-1} c(g|k) f_k(\underline{x}) p_k}_{\text{what we pay if we put } \underline{x} \text{ in g and it's not}} d\underline{x}$$

$\Rightarrow$ we'll put $\underline{x}$ where we have to pay less if we've wrong !!

We want to find $R_1, \dots, R_g$ which minimize $ECM(\delta)$

$$R_1 = \{ \underline{x} \in \mathbb{R}^p : \underbrace{\sum_{k=2}^{g} c(1|k) f_k(\underline{x}) p_k \leq \sum_{k \neq j} c(j|k) f_k(\underline{x}) p_k}_{\text{cost}\left(\begin{matrix}\underline{x} \text{ is } k \ (k=2,\dots,g) \\ \text{we put } 1\end{matrix}\right) \leq \text{cost}\left(\begin{matrix}\underline{x} \text{ is } j \ (j \neq k) \\ \text{we put } k\end{matrix}\right) \ \forall j \neq 1}, \ j = 2, 3, \dots, g \}$$

$$R_2 = \{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq 2} c(2|k) f_k(\underline{x}) p_k \leq \sum_{k \neq j} c(j|k) f_k(\underline{x}) p_k, \ j = 1, 3, \dots, g \}$$

$$R_i = \{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} c(i|k) f_k(\underline{x}) p_k \leq \sum_{k \neq j} c(j|k) f_k(\underline{x}) p_k, \ j \neq i \}$$

$\longrightarrow$ optimal classifier (in the sense of min ECM): $\delta(\underline{x}) = i \iff \underline{x} \in R_i$

Note:

$$\delta(\underline{x}) = t \quad \Longleftrightarrow \quad \left[ \sum_{k \neq t} c(t|k) f_k(\underline{x}) p_k \leq \sum_{k \neq j} c(j|k) f_k(\underline{x}) p_k \right] \qquad (*)$$

$t \in \{1, \dots, g\}$

$$\Longrightarrow \quad \frac{\sum_{k \neq t} c(t|k) f_k(\underline{x}) p_k}{\sum_{k=1}^{g} f_k(\underline{x}) p_k} \leq \frac{\sum_{k \neq j} c(j|k) f_k(\underline{x}) p_k}{\sum_{k=1}^{g} f_k(\underline{x}) p_k} \qquad ((*))$$

$*$

(by def.) $= \sum_{k=1}^{g} \mathbb{P}(\underline{X} = \underline{x} \mid L = k) \, \mathbb{P}(L = k) = \mathbb{P}(\underline{X} = \underline{x})$

$* \neq 0$ $(> 0)$ because if all the densities are zero in that $\underline{x}$ then we'll never see that $\underline{x}$

Moreover:

$$\frac{f_k(\underline{x}) p_k}{\sum_{k=1}^{g} f_k(\underline{x}) p_k} = \frac{\mathbb{P}(\underline{X} = \underline{x} \mid L = k) \, \mathbb{P}(L = k)}{\mathbb{P}(\underline{X} = \underline{x})} = \frac{\mathbb{P}(\underline{X} = \underline{x}, L = k)}{\mathbb{P}(\underline{X} = \underline{x})}$$

$$= \mathbb{P}(L = k \mid \underline{X} = \underline{x}) \qquad \text{(BAYES)}$$

$\Longrightarrow$ $(*)$ becomes:

OPTIMAL CLASSIFIER

$$\delta(\underline{x}) = t \quad \Longleftrightarrow \quad \sum_{k \neq t} c(t|k) \, \mathbb{P}(L = k \mid \underline{X} = \underline{x}) \leq \sum_{k \neq j} c(j|k) \, \mathbb{P}(L = k \mid \underline{X} = \underline{x})$$

the expected posterior cost $\leq$ all the other expected cost for all the other groups

- BIG Assumption: $c(i|j) = \text{const} > 0$ for $i \neq j$ (all costs are the same)

$\Longrightarrow$ $(*)$ becomes:

$$\delta(\underline{x}) = t \quad \Longleftrightarrow \quad \sum_{k \neq t} \mathbb{P}(L = k \mid \underline{X} = \underline{x}) \leq \sum_{k \neq j} \mathbb{P}(L = k \mid \underline{X} = \underline{x})$$

$$\Longleftrightarrow \quad 1 - \mathbb{P}(L = t \mid \underline{X} = \underline{x}) \leq 1 - \mathbb{P}(L = j \mid \underline{X} = \underline{x})$$

$$\Longleftrightarrow \quad \mathbb{P}(L = j \mid \underline{X} = \underline{x}) \leq \mathbb{P}(L = t \mid \underline{X} = \underline{x})$$

if we're assuming that all the costs of misclassification are the same we end up with an optimal classifier that says: "we attribute to $\underline{x}$ the label $t$ if the posterior probability of belonging to group $t$ is maximum" := BAYES CLASSIFIER

**BAYES CLASSIFIER**

special case of the optimal (general) classifier. Special case because we're assuming the costs to be the same.

- **BIG Assumption** : $c(i|j) = \text{const}$ for $i \neq j$ and $p_1 = p_2 = \dots = p_g = \frac{1}{g}$

$$\Rightarrow \mathbb{P}(L=j \mid \underline{X}=\underline{x}) \leq \mathbb{P}(L=t \mid \underline{X}=\underline{x})$$

$$\rightarrow \frac{f_j(\underline{x})\, p_j}{\mathbb{P}(\underline{X}=\underline{x})} \leq \frac{f_t(\underline{x})\, p_t}{\mathbb{P}(\underline{X}=\underline{x})} \rightarrow \boxed{f_j(\underline{x}) \leq f_t(\underline{x})}$$

(MAXIMUM LIKELIHOOD)

**MLE** CLASSIFIER

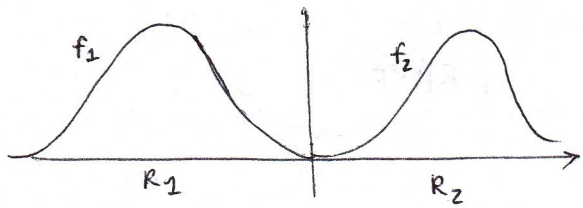
the classifier will attribute the unit for which we have observed $\underline{x}$ to $t$ if the density $f_t(\underline{x})$ is larger than all the other densities.

We attribute to the group for which the likelihood of what we have observed is maximized.

Example: $g=2$, $p=1$

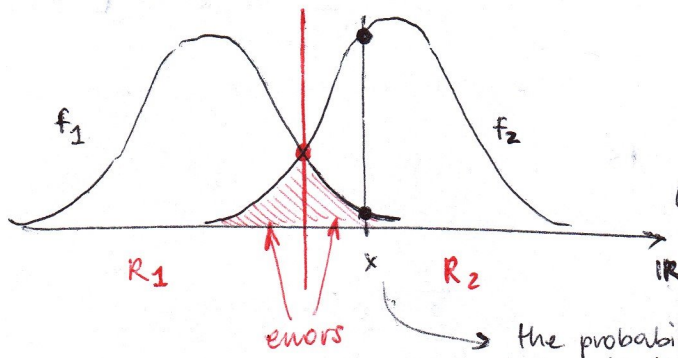
We observe how tall people are ($g=2 \Rightarrow$ male, females)

We can have this situation:




but it's not interesting, it's like features that are shown by units belonging to group 1 are not showned by units belonging to group 2. (and viceversa).

Another case:




**MLE CLASSIFIER**

(heights)

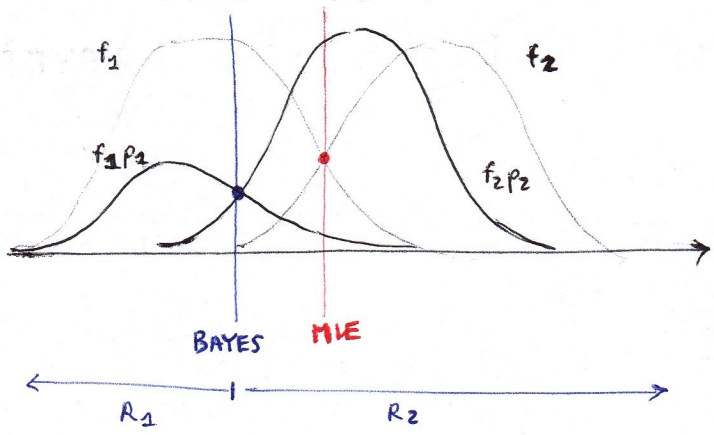
→ the probability of observing $x$ with that height is longer for group 2 than group 1

$\Rightarrow x \in R_2$

If we have a prior : suppose # males >> # females :

$f_1 \rightarrow f_1 p_1$

$f_2 \rightarrow f_2 p_2$




the Bayes classifier attributes to group 2 or 1 according to the posterior distribution $(f_1 p_1, f_2 p_2)$

(Bayes: more conservative)

Optimal classifier: (min ECM)

$$\delta(\underline{x}) = i \quad \text{if} \quad \underline{x} \in R_i \qquad i = 1, \dots, g$$

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} c(i|k) f_k(\underline{x}) p_k \leq \sum_{k \neq j} c(j|k) f_k(\underline{x}) p_k \;,\; j = 1, \dots, g \right\}$$

$$= \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} c(i|k) \mathbb{P}(L = k | \underline{X} = \underline{x}) \leq \sum_{k \neq j} c(j|k) \mathbb{P}(L = k | \underline{X} = \underline{x}) \;,\; j = 1, \dots, g \right\}$$

Observations:

- costs could be specified up to a multiplicative constant $> 0$ (euro $\leftrightarrow$ dollars $\Rightarrow$ same classifier)
- It's a **Bayesian** classifier (because we're computing the posterior probabilities and then we're computing the expected cost w.r.t. these posterior probabilities. So, it's not that if we are not using "THE" Bayes classifier then it's not Bayesian.)

Let's get deeper on the **BAYES CLASSIFIER** (not the generical one)

If $c(i|j) = \text{constant} > 0$ for $i \neq j \Rightarrow$ Bayes classifier, i.e.

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \mathbb{P}(L = i | \underline{X} = \underline{x}) \geq \mathbb{P}(L = j | \underline{X} = \underline{x}) \;,\; j = 1, \dots, g \right\}$$

it seems like it forgets about the costs and looks only at posterior probabilities and attributes a unit for which we observed $\underline{x}$ to the group which has max post. prob.

Bayes classifier is more "flexible" than what it seems.

For instance, consider the optimal classifier and assume that:

$$c(i|k) = c_k \geq 0 \qquad i, k = 1, \dots, g$$

constant in every group, **not** all over: whenever we make a mistake in classifing a unit which belong to group $k$ we pay a fixed cost (doesn't matter if we're attributing it at 1, 2, ...)

$\Rightarrow$ the optimal classifier is:

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} f_k(\underline{x}) c_k p_k \leq \sum_{k \neq j} f_k(\underline{x}) c_k p_k \qquad j = 1, \dots, g \right\}$$

$$= \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} f_k(\underline{x}) \frac{c_k p_k}{\sum c_j p_j} \leq \sum_{k \neq j} f_k(\underline{x}) \frac{c_k p_k}{\sum c_j p_j} \qquad j = 1, \dots, g \right\}$$

Let $\pi_k := \dfrac{c_k p_k}{\sum c_j p_j} \qquad k = 1, \dots, g$

Note that $\pi_k \geq 0$ and $\sum \pi_k = 1 \Rightarrow \pi_k$ are acting like priors distributions

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{k \neq i} f_k(\underline{x}) \pi_k \leq \sum_{k \neq j} f_k(\underline{x}) \pi_k \qquad j = 1, \dots, g \right\} \Rightarrow \text{BAYES CLASSIFIER with priors } \pi_k$$

This is Bayes with priors $\pi_1, \dots, \pi_g$.

We have costs, but we're modifing the priors to take in account the costs and then we're back to the Bayes classifier (costs and priors play a similar role)

Exercise: work out the optimal classifier when

- $c(i|k) = c_i \geq 0$ for $i, k = 1, \dots, g$
- $c(i|k) = c_i \cdot h_k$ for $i, k = 1, \dots, g$
- $c(i|k) = c^{ai} \cdot b^{bk}$

} the optimal classifier is just a slightly modification of the Bayes classifier

## Special cases of Bayes classifiers (2)

Assume that $\underline{X} \mid L=i \sim N_p(\mu_i, \Sigma_i)$ $\quad i=1,\dots,g$ (the distribution of the features is gaussian)

$$\Rightarrow \quad \mathbb{P}(L=i \mid \underline{X}=\underline{x}) \geq \mathbb{P}(L=j \mid \underline{X}=\underline{x})$$

$$\Rightarrow \quad \frac{f_i(\underline{x})\, p_i}{\mathbb{P}(\underline{X}=\underline{x})} \geq \frac{f_j(\underline{x})\, p_j}{\mathbb{P}(\underline{X}=\underline{x})} \quad \Rightarrow \quad f_i(\underline{x})\, p_i \geq f_j(\underline{x})\, p_j$$

$$\frac{p_i}{\sqrt{(2\pi)^p \, \mathrm{Det}(\Sigma_i)}} e^{-\frac{1}{2}(\underline{x}-\mu_i)^T \Sigma_i^{-1} (\underline{x}-\mu_i)} \geq \frac{p_j}{\sqrt{(2\pi)^p \, \mathrm{Det}(\Sigma_j)}} e^{-\frac{1}{2}(\underline{x}-\mu)^T \Sigma_j^{-1} (\underline{x}-\mu)}$$

Taking the log:

$$\log(p_i) - \frac{1}{2}\log\left(\mathrm{Det}(\Sigma_i)\right) - \frac{1}{2}(\underline{x}-\mu_i)^T \Sigma_i^{-1} (\underline{x}-\mu_i)$$

$$\geq \log(p_i) - \frac{1}{2}\log\left(\mathrm{Det}(\Sigma_j)\right) - \frac{1}{2}(\underline{x}-\mu_j)^T \Sigma_j^{-1} (\underline{x}-\mu_j)$$

Def. $d_i^Q : \mathbb{R}^p \to \mathbb{R}$,

$$d_i^Q(\underline{x}) = \log(p_i) - \frac{1}{2}\log\left(\mathrm{Det}(\Sigma_i)\right) - \frac{1}{2}(\underline{x}-\mu_i)^T \Sigma_i^{-1} (\underline{x}-\mu_i) \qquad i=1,\dots,g$$

QUADRATIC DISCRIMINANT SCORE FUNCTIONS

quadratic part, in fact this is the Mahalanobis' distance between $\underline{x}$ and $\mu_i$: $= d^2_{\Sigma_i^{-1}}(\underline{x}, \mu_i)$

$\Rightarrow$ the optimal classifier becomes:

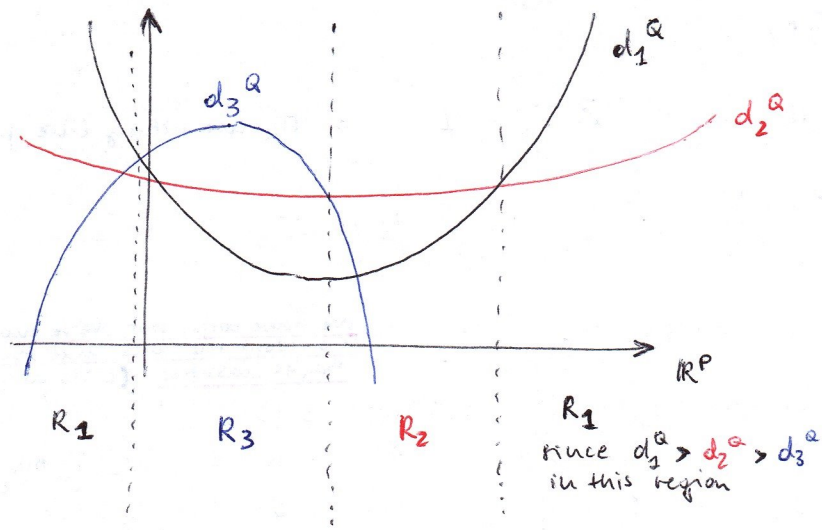
$$\delta(\underline{x}) = i \quad \text{if } \underline{x} \in R_i \qquad i=1,\dots,g$$

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \; d_i^Q(\underline{x}) \geq d_j^Q(\underline{x}) \;,\; j=1,\dots,g \right\}$$

$\Rightarrow$ the closer we are to the mean $\mu_i$, the higher is the chance that we belong to group $i$ (Mahalanobis' "closedness")

**(1) QDA** = QUADRATIC DISCRIMINANT ANALYSIS = gaussian model for the distribution of the features in each group + Bayes classifier (costs equal or modified version with special structure for the costs like $c(i|k) = c_k, \dots$)

Example:



Assume moreover that : $\Sigma_1 = \Sigma_2 = \dots = \Sigma_g = \Sigma$

then:

$d_i^Q(\underline{x}) \geq d_j^Q(\underline{x})$ equals to:

$$\log(p_i) - \frac{1}{2}\log(\text{Det}(\Sigma)) - \frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma^{-1}(\underline{x}-\underline{\mu}_i) \geq \log(p_j) - \frac{1}{2}\log(\text{Det}(\Sigma)) - \frac{1}{2}(\underline{x}-\underline{\mu}_j)^T\Sigma^{-1}(\underline{x}-\underline{\mu}_j) \quad (*)$$

note that:

$$-\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T\Sigma^{-1}(\underline{x}-\underline{\mu}_i) = -\frac{1}{2}\underline{x}^T\Sigma^{-1}\underline{x} + \underline{\mu}_i^T\Sigma^{-1}\underline{x} - \frac{1}{2}\underline{\mu}_i^T\Sigma^{-1}\underline{\mu}_i$$

$\Rightarrow$ (*) becomes:

$$\log(p_i) + \underline{\mu}_i^T\Sigma^{-1}\underline{x} - \frac{1}{2}\underline{\mu}_i\Sigma^{-1}\underline{\mu}_i \geq \log(p_j) + \underline{\mu}_j^T\Sigma^{-1}\underline{x} - \frac{1}{2}\underline{\mu}_j^T\Sigma^{-1}\underline{\mu}_j$$

Def. $d_i : \mathbb{R}^p \to \mathbb{R}$

$$d_i(\underline{x}) = \log(p_i) + \underline{\mu}_i^T\Sigma^{-1}\underline{x} - \frac{1}{2}\underline{\mu}_i^T\Sigma\underline{\mu}_i$$

**LINEAR DISCRIMINANT SCORE FUNCTIONS** (linear in $\underline{x}$)

- features are gaussian
- covariances are all the same in each group

$\Rightarrow$ the optimal classifier becomes:

$$\delta(\underline{x}) = i \quad \text{if } \underline{x} \in R_i \qquad i = 1, \dots, g$$

$$R_i = \{\underline{x} \in \mathbb{R}^p : d_i(\underline{x}) \geq d_j(\underline{x}), \quad j = 1, \dots, g\}$$

(*) This is important: the computer will try to estimate priors as the proportions of the training dataset. We have to modify it. Remember, btw, that we're not trying to get the right priors, the **goal** is to have a good classifier for the labels. (they're 2 ≠ problems)

(2) **LDA** = LINEAR DISCRIMINANT ANALYSIS

We've done with general classifiers.
We worked only on the model, now we use the training data to estimate all the parameters that enter in this model. We'll use the training set to estimate the means, the covariances if we're working with LDA / QDA, otherwise we'll use the training set to estimate the distribution in different groups.
**NOTE**: do not use the training set to estimate the PRIORS PROBABILITIES (unless we have a training set that is a sample (PERFECT SAMPLE) of the real population with the right proportions, i.e. 10% in all the population is group 1 $\Rightarrow$ 10% in the sample is group 1. This is usually not the case of training sets).

## ESTIMATE THE PARAMETERS

We estimate the parameters using the training set

$$\underset{\text{training set}}{\mathbb{X}} = \begin{bmatrix} \underline{x}_1^T & l_1 \\ \vdots & \vdots \\ \underline{x}_n^T & l_n \end{bmatrix} \qquad \underline{x}_i \in \mathbb{R}^p \qquad l_i \in \{1, \dots, g\}$$

$p_k$ $k = 1, \dots, g$: probability of an individual to belong to a group $k$ before we see the $\underline{x}$ vector of features of the individual

- For **QDA**:

$$\hat{\underline{\mu}}_k = \frac{1}{n_k}\sum_{\{i : l_i = k\}} \underline{x}_i = \bar{\underline{x}}_k \qquad n_k = \#\{i \in \{1, \dots, n\} : l_i = k\}$$

$$\hat{\Sigma}_k = \frac{1}{n_k - 1}\sum_{\{i : l_i = k\}} (\underline{x}_i - \bar{\underline{x}}_k)(\underline{x}_i - \bar{\underline{x}}_k)^T = S_k$$

- For **LDA**:

$$\hat{\mu}_k = \bar{X}_k$$

$$\hat{\Sigma} = \frac{1}{n-g} \sum_{k=1}^{g} S_k (n_k - 1)$$ pooled estimator of the covariance

Problem when $p$ is very large w.r.t. $n$, or many missing data in the features $\Longrightarrow$ difficult (impossible) to compute $S_i^{-1}$

$\longrightarrow$ assume a model (parametric) for $\Sigma_i$'s (we reduce the dimensionality of the problem)

For instance: (Example!)

Assume that in each group the components of $\underline{X}$ are independent, i.e.

$$\Sigma_k = \begin{bmatrix} \sigma_{11}^{(k)} & & \emptyset \\ & \ddots & \\ \emptyset & & \sigma_{pp}^{(k)} \end{bmatrix}$$

then, for $k = 1, \dots, g$:

$$d_k^Q(\underline{x}) = \log(p_k) - \frac{1}{2} \sum_{j=1}^{p} \log\left(\sigma_{jj}^{(k)}\right) - \frac{1}{2} \sum_{j=1}^{p} (x_j - \mu_{kj})^2 \frac{1}{\sigma_{jj}^{(k)}}$$

$\mu_{kj}$ : $j$-th component of $\mu_k$

we need to estimate:

$$\hat{\mu}_k = \bar{\underline{x}}_k \qquad k = 1, \dots, g$$

$$\sigma_{jj}^{(k)} = \frac{1}{n_k - 1} \sum_{\{i : l_i = k\}} (x_{ij} - \bar{X}_{kj})^2$$

(do-able even if $p > n_k$)

we need at least 2 observations in each group for each component

**NAIVE BAYES GAUSSIAN CLASSIFIER**

because we're assuming independence

$\Downarrow$

we need a model just for univariate distribution (since they're $\perp\!\!\!\perp$)

## EVALUATING A CLASSIFIER by ESTIMATING ITS ERROR RATE

$\delta$ is a classifier

$\delta : \mathbb{R}^p \longrightarrow \{1, \dots, g\}$

Actual error rate of $\delta$: $AER(\delta)$

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

$$+ \underbrace{\sum_{k \neq g} \int_{R_k} f_g(\underline{x}) p_g \, d\underline{x}}$$

NOTE: we cannot compute it because we don't know the right values of the priors and densities $\Rightarrow$ ESTIMATES

!! With probability $p_g$ a unit belongs to group $g$ and it'll show a feature $\underline{x}$ with probability $f_g(\underline{x})$. It'll be wrongly classified if it's attributed to a group $k \neq g$.

Estimate **AER** **non**-parametrically : ⇒ without using a model

to make it easy assume that $g = 2$

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

Compute the **CONFUSION MATRIX** by applying $\delta$ to the training set

| Actual L \ Attributed L | 1 | 2 |
|---|---|---|
| 1 | $n_{11}$ | $n_{12}$ |
| 2 | $n_{21}$ | $n_{22}$ |

we can compute this since we're doing the count of errors on the training set ⇒ we know the true labels

$$\text{APER}(\delta) = \frac{n_{12} + n_{21}}{n} = \frac{\#\text{mistakes}}{\#\text{trials}} \longrightarrow$$

APPARENT ERROR RATE

too optimistic (we need to test the classifier on a NEW dataset)

$$\frac{n_{12} + n_{21}}{n} = \frac{n_1}{n}\frac{n_{12}}{n_1} + \frac{n_2}{n}\frac{n_{21}}{n}$$

$$= \hat{p}_1 \left( \int_{R_2} \hat{f}_1(\underline{x})\, d\underline{x} \right) + \hat{p}_2 \left( \int_{R_1} \hat{f}_2(\underline{x})\, d\underline{x} \right)$$

Better estimate of $AER(\delta)$ : we want to use the dataset for both learning and testing but we don't want to test the classifier with the same data that we used to build it

⟹ **Leave-one-out** cross-validation

For $i = 1, \dots, n$

1. We take the unit $i$ "out" of the training set

$$\Rightarrow \quad \underline{\underline{X}}_{-i} = \begin{bmatrix} \underline{x}_1^T\, l_1 \\ \vdots \\ \underline{x}_{i-1}^T\, l_{i-1} \\ \underline{x}_{i+1}^T\, l_{i+1} \\ \vdots \\ \underline{x}_n^T\, l_n \end{bmatrix}$$

2. We train $\delta$ on $\underline{\underline{X}}_{-i} \Rightarrow \delta_{-i} : \mathbb{R}^p \longrightarrow \{1, \dots, g\}$

($\delta_{-i}$ is similar to the trained $\delta$) since the difference between the two datasets they're trained on is of **1** unit

3. We apply $\delta_{-i}$ to $\underline{x}_i \Rightarrow \delta_{-i}(\underline{x}_i) = \hat{l}_i$

4. $\varepsilon_i = \begin{cases} 1 & \hat{l}_i \neq l_i \\ 0 & \hat{l}_i = l_i \end{cases}$

$$\Rightarrow \quad \hat{AER}(\delta) = \frac{\sum_{i=1}^{n} \varepsilon_i}{n}$$

**L10** estimate of $AER(\delta)$

(**Note**: in the end we take $\delta$, not some $\delta_{-i}$ !)

Better idea: **K- FOLD** cross validation (L10 = 1-fold cross validation)

(L10 is unbiased (most of the time), but there's too much variability : if we change the training set a little bit we'll have a different estimate ⇒ **LKO** "Leave k-out")

# CROSS VALIDATION

24/04

L10 cross-validation $\Longrightarrow$ $\hat{AER}(\delta)$ : small bias, high variance

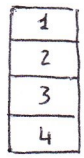
we are very uncertain about how close this estimate is to the true actual error rate

How to reduce the variance without sacrifying too much of the bias?
Answer: try k-fold cross validation.

there is always a bias-variance trade-off, the prediction error is the sum of the two.

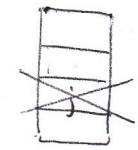
## k-fold cross validation algorithm:

0. Set $k < n$ (usually $k = 5, 10$ for $n$ large enough, and randomly split the units of the training set in $k$ parts.

$\mathbb{X}$ = 


randomly means : permute first the rows of $\mathbb{X}$ and then split in k parts (n! permutations, we can't do all of them, we have to chose one)

For $j = 1, \dots, k$ :

1. Hold out part $j$ from the training set $\Longrightarrow$ $\mathbb{X}_{-\text{part}\,j}$ = 

2. Train $\delta$ on $\mathbb{X}_{-\text{part}\,j}$ :
$\delta_{-\text{part}\,j} : \mathbb{R}^p \longrightarrow \{1, \dots, g\}$

3. Apply $\delta_{-\text{part}\,j}$ to part $j$ and count the errors :

$$Err_j = \frac{1}{n_j} \sum_{\{i \in \text{part}\,j\}} \varepsilon_i$$

$$n_j = \#\{i \in \{1, \dots, n\} : i \in \text{part}\,j\}$$

$$\varepsilon_i = \begin{cases} 1 & \delta_{-\text{part}\,j}(\underline{x}_i) \neq l_i \\ 0 & \delta_{-\text{part}\,j}(\underline{x}_i) = l_i \end{cases}$$

4. $\hat{AER}(\delta) = \frac{1}{n} \sum_{j=1}^{k} n_j \, Err_j$

Obs. If $k = n$ $\Longrightarrow$ L10

Obs. By initializing k-fold B times, each time selecting a permutation at random of the rows of $\mathbb{X}$ before splitting ;

$\Longrightarrow$ $\hat{AER}_1(\delta), \dots, \hat{AER}_B(\delta)$

$\Longrightarrow$ we can compute the mean:

$$\hat{AER}_m(\delta) = \frac{1}{B} \sum_{j=1}^{B} \hat{AER}_j(\delta)$$

(estimate of $\mathbb{E}[\hat{AER}(\delta)]$)

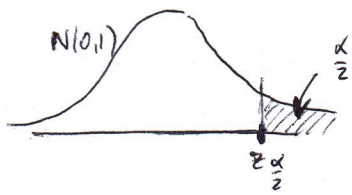
and also the variance:

$$Var\left(\hat{AER}(\delta)\right) = \frac{1}{B-1}\sum_{j=1}^{B}\left(\hat{AER}_j(\delta) - \hat{AER}_m(\delta)\right)^2$$

$\Rightarrow$ we can compute:

$$CI_{1-\alpha}\left(\mathbb{E}[\hat{AER}(\delta)]\right) = \left[\hat{AER}_m(\delta) \pm \sqrt{\frac{Var(\hat{AER}(\delta))}{B}} \cdot z_{\frac{\alpha}{2}}\right] \quad (CLT)$$

we can't do it with "Leave-1-out" since we have only 1 permutation
$\Rightarrow$ k-fold is more flexible




## Why k-folds cross validation reduces the variability?

L10 is based on the average of: $\delta_{-1}, \delta_{-2}, \dots, \delta_{-n}$.

they're all based on the same dataset but for 1 observation $\Rightarrow$ they're almost copies one of another $\Rightarrow$ if we take the average the variability is not reduced (taking average reduce variability with II observations)

They're strongly correlated: (II)

$$\Rightarrow Var\left(\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\right) \simeq Var(\varepsilon_j)$$

if data are strongly correlated it's like having many times the same element $\Rightarrow$ the variance in one elem. its equal to the variance of the mean

k-fold cross-validation averages: $\delta_{-part\,1}, \delta_{-part\,2}, \dots, \delta_{-part\,k}$

are less correlated since their datasets (from which they're trained) have more than 1 observation different

could these are less correlated:

$$\Rightarrow Var\left(\frac{1}{n}\sum_{j=1}^{k} n_j Err_j\right) < Var(Err_j)$$

There is a high chance that:

$$Var\left(\frac{1}{n}\sum_{j=1}^{k} n_j Err_j\right) < Var\left(\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\right)$$

at the cost of higher bias.

---

(Long) observation on LDA:

**LDA is robust with the Gaussianity assumption**
**(Fisher's argument for LDA)**

$$\underline{X} \mid L = i \sim (\underline{\mu}_i, \Sigma) \quad \text{(we've not assuming gaussianity)} \quad i = 1, \dots, g$$

$\hookrightarrow$ same for all groups

For $\underline{a} \in \mathbb{R}^p$:

$$\mathbb{E}[\underline{a}^T\underline{X} \mid L = i] = \underline{a}^T\underline{\mu}_i \Rightarrow$$ the mean of every group along the direction $\underline{a}$ is different

$$Var(\underline{a}^T\underline{X} \mid L = i] = \underline{a}^T\Sigma\,\underline{a} \quad \forall i = 1, \dots, g \Rightarrow$$ same variability along direction $\underline{a}$ for every group $i$

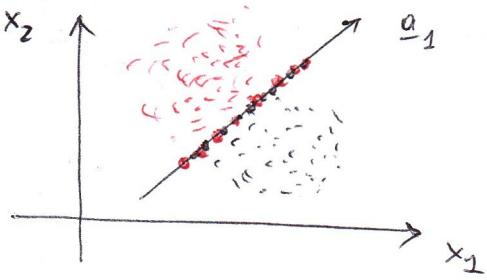
**Goal**: find $\underline{a}$ (a direction) which maximize the variability **BETWEEN** groups w.r.t. the variability **WITHIN** groups

Covariability between groups:

$$B = \frac{1}{g-1}\sum_{i=1}^{g}(\underline{\mu}_i - \bar{\underline{\mu}})(\underline{\mu}_i - \bar{\underline{\mu}})^T, \qquad \bar{\underline{\mu}} = \frac{1}{g}\sum_{i=1}^{g}\underline{\mu}_i$$

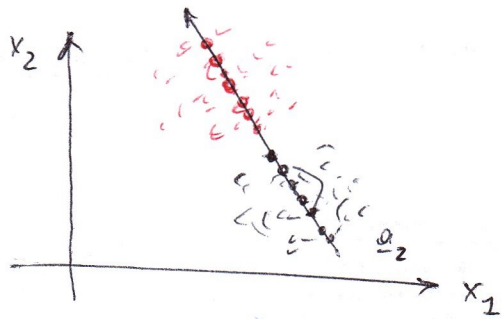
Covariance within groups: $\Sigma$

**Idea:** find $\underline{a} \in \mathbb{R}^p$ which maximizes the separation between groups.




this direction ($\underline{a}_1$) does not separate the two groups: the two groups projected on this direction are totally confounded

(two features)




this direction ($\underline{a}_2$) separates the groups: we can say "these above the treshold are red and these below are black"

⟹ we identified a sort of linear separation between the two groups

How to find this direction $\underline{a}$?

$$\Longrightarrow \quad \arg\max_{\underline{a} \in \mathbb{R}^p} \frac{\underline{a}^T B \underline{a}}{\underline{a}^T \Sigma \underline{a}} = \arg\max_{\underline{a} \in \mathbb{R}^p} \frac{1}{g-1} \frac{\sum (\underline{a}^T \underline{\mu}_i - \underline{a}^T \underline{\bar{\mu}})^2}{\underline{a}^T \Sigma \underline{a}}$$

We re-parametrize:

$$\underline{u} := \Sigma^{1/2} \underline{a} \longrightarrow \underline{a} = \Sigma^{-1/2} \underline{u}$$

$$\Longrightarrow \frac{\underline{a}^T B \underline{a}}{\underline{a}^T \Sigma \underline{a}} = \frac{\underline{u}^T \Sigma^{-1/2} B \Sigma^{-1/2} \underline{u}}{\underline{u}^T \Sigma^{-1/2} \Sigma \Sigma^{-1/2} \underline{u}} = \frac{\underline{u}^T \Sigma^{-1/2} B \Sigma^{-1/2} \underline{u}}{\underline{u}^T \underline{u}}$$

if $\Sigma^{-1/2} B \Sigma^{-1/2} = \sum_{i=1}^{s} \lambda_i \underline{e}_i \underline{e}_i^T$ $\qquad$ $\underline{s} = \min(p, g-1)$ = number of eigenvalues $\neq 0$

- $p$ = order of $\Sigma$: we have $p$ components for the features
- $g-1$ = degrees of freedom for $B$ (since we're comparing a covariance with $g$ data)

$$\Longrightarrow \arg\max_{\underline{u}} \frac{\underline{u}^T \Sigma^{-1/2} B \Sigma^{-1/2} \underline{u}}{\underline{u}^T \underline{u}} = \underline{e}_1$$

$$\Longrightarrow \arg\max_{\underline{a}} \frac{\underline{a}^T B \underline{a}}{\underline{a}^T \Sigma \underline{a}} = \Sigma^{-1/2} \underline{e}_1$$

Following the convention: $\lambda_1 \geq \lambda_2 \geq \dots$

As in PCA: $\underline{a}_2 = \Sigma^{-1/2} \underline{e}_2, \dots, \underline{a}_s = \Sigma^{-1/2} \underline{e}_s$ → best discriminating directions after we have considered $\underline{a}_1$ and so on

$$A = \begin{bmatrix} \underline{a}_1^T \\ \vdots \\ \underline{a}_s^T \end{bmatrix} \Longrightarrow \boxed{\mathrm{Cov}(A\underline{X}) = I} \quad (*)$$

not only we have found $s$ directions along which we maximize the separation but also they're all non correlated ⇒ the scores on these $s$ directions are not correlated

(*) $\mathrm{Cov}(\underline{a}_i \underline{X}, \underline{a}_j \underline{X}) = \underline{a}_i^T \Sigma \underline{a}_j$

$$= \underline{e}_i^T \Sigma^{-1/2} \Sigma \Sigma^{-1/2} \underline{e}_j = \underline{e}_i \underline{e}_j^T = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

$\underline{a}_1^T \underline{X}, \dots, \underline{a}_s^T \underline{X}$ := first, second, ..., ... **Fisher's discriminant scores**

since $\mu_1, \dots, \mu_g$ and $\Sigma$ are unknown $\Rightarrow$ we estimate them using training data

$$\rightarrow \begin{cases} \hat{\mu}_i = \bar{\underline{X}}_i \\ \hat{\Sigma} = \frac{1}{n-g} \sum_{i=1}^{g} (n_i - 1) S_i \end{cases}$$

**Note that**: we can use Fisher's discriminant scores not only for classifing objects but also for dimension reduction.

Building a classifier by means of Fisher's scores: (3 steps)

1. $\bar{\underline{X}}_i \longrightarrow$ we project the means on $\underline{a}_1, \dots, \underline{a}_s$

$$\begin{bmatrix} \underline{a}_1^T \bar{\underline{X}}_i \\ \underline{a}_2^T \bar{\underline{X}}_i \\ \vdots \\ \underline{a}_s^T \bar{\underline{X}}_i \end{bmatrix} \quad i = 1, \dots, g$$

we consider just the first k - projections (we decide to reduce dimension to k)

To classify the unit for which we have observed $\underline{x}$:

2. $\underline{x} \longrightarrow \begin{bmatrix} \underline{a}_1^T \underline{x} \\ \vdots \\ \underline{a}_k^T \underline{x} \end{bmatrix}$ we project $\underline{x}$ on the space "formed" by $\underline{a}_1, \dots, \underline{a}_k$ selected before

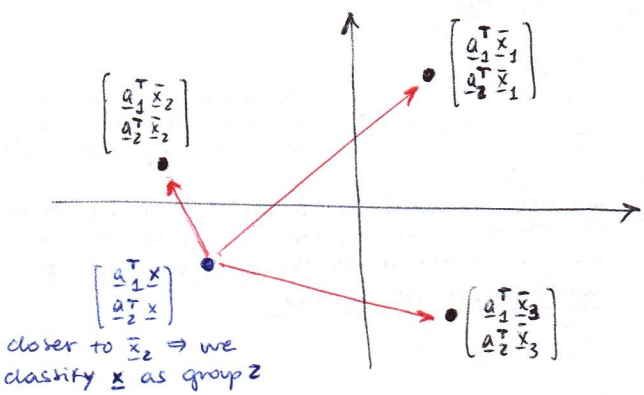
3. attribute $\underline{x}$ to the closest mean $\left( \begin{bmatrix} \underline{a}_1^T \bar{\underline{X}}_i \\ \vdots \\ \underline{a}_s^T \bar{\underline{X}}_i \end{bmatrix} \quad i = 1, \dots, g \right)$

more precisely:

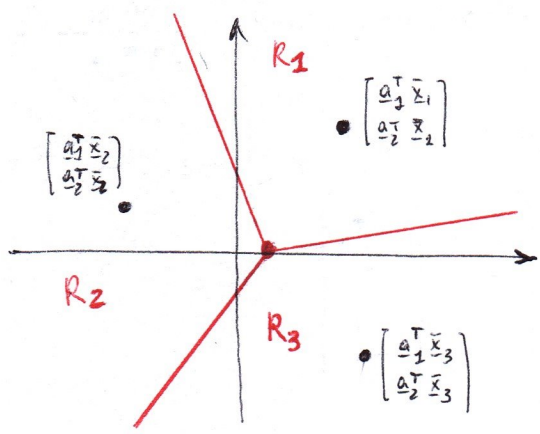
Note that the covariance is the identity matrix, so the statistical distance in this space is the euclidean distance

$$\delta(\underline{x}) = i \quad \text{if } \underline{x} \in R_i \qquad i = 1, \dots, g$$

$$R_i = \left\{ \underline{x} \in \mathbb{R}^p : \sum_{j=1}^{k} (\underline{a}_j^T \underline{x} - \underline{a}_j^T \bar{\underline{X}}_i)^2 \leq \sum_{j=1}^{k} (\underline{a}_j^T \underline{x} - \underline{a}_j^T \bar{\underline{X}}_h)^2 \quad h = 1, \dots, g \right\}$$




doing it $\forall \underline{x}$ $\Rightarrow$




This classifier is the same as LDA when the priors are the same.

$(p_1 = p_2 = \dots = p_g = \frac{1}{g})$

(the boundaries of the regions are linear)

## Classification

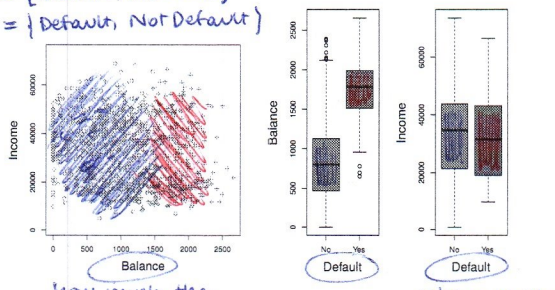
- Qualitative variables take values in an unordered set $\mathcal{C}$, such as:
  `eye color` $\in$ {`brown`, `blue`, `green`}
  `email` $\in$ {`spam`, `ham`}.
- Given a feature vector $X$ and a qualitative response $Y$ taking values in the set $\mathcal{C}$, the classification task is to build a function $C(X)$ that takes as input the feature vector $X$ and predicts its value for $Y$; i.e. $C(X) \in \mathcal{C}$.
- Often we are more interested in estimating the *probabilities* that $X$ belongs to each category in $\mathcal{C}$.

  For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification fraudulent or not.



## Example: Credit Card Default

$\underline{X} = (\text{income}, \text{balance})^T$
$L = \{\text{Default}, \text{Not Default}\}$




How much the individual own to the bank at the end of the month (paying back the debt from credit card)

= bancarotta



## Can we use Linear Regression?

Suppose for the `Default` classification task that we code

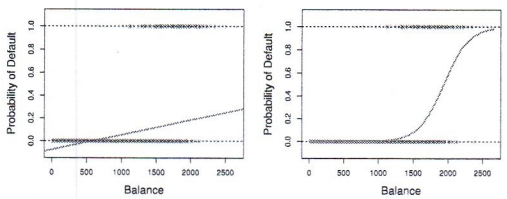
$$Y = \begin{cases} 0 & \text{if } \texttt{No} \\ 1 & \text{if } \texttt{Yes}. \end{cases}$$

Can we simply perform a linear regression of $Y$ on $X$ and classify as `Yes` if $\hat{Y} > 0.5$?

- In this case of a binary outcome, linear regression does a good job as a classifier, and is equivalent to *linear discriminant analysis* which we discuss later.
- Since in the population $E(Y|X = x) = \Pr(Y = 1|X = x)$, we might think that regression is perfect for this task.
- However, *linear* regression might produce probabilities less than zero or bigger than one. *Logistic regression* is more appropriate.



## Linear versus Logistic Regression




The orange marks indicate the response $Y$, either 0 or 1. Linear regression does not estimate $\Pr(Y = 1|X)$ well. Logistic regression seems well suited to the task.



## Linear Regression continued

Now suppose we have a response variable with three possible values. A patient presents at the emergency room, and we must classify them according to their symptoms.

$$Y = \begin{cases} 1 & \text{if } \texttt{stroke}; \\ 2 & \text{if } \texttt{drug overdose}; \\ 3 & \text{if } \texttt{epileptic seizure}. \end{cases}$$

This coding suggests an ordering, and in fact implies that the difference between `stroke` and `drug overdose` is the same as between `drug overdose` and `epileptic seizure`.

Linear regression is not appropriate here.
*Multiclass Logistic Regression* or *Discriminant Analysis* are more appropriate.



## Logistic Regression

Let's write $p(X) = \Pr(Y = 1|X)$ for short and consider using `balance` to predict `default`. Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

($e \approx 2.71828$ is a mathematical constant [Euler's number.])
It is easy to see that no matter what values $\beta_0$, $\beta_1$ or $X$ take, $p(X)$ will have values between 0 and 1.

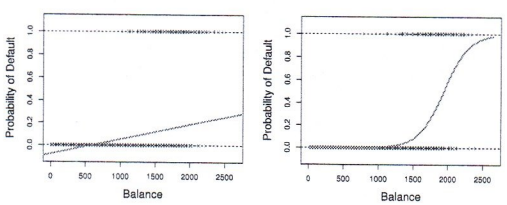
A bit of rearrangement gives

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X.$$

This monotone transformation is called the *log odds* or *logit* transformation of $p(X)$. (by log we mean *natural log*: ln.)



## Linear versus Logistic Regression




Logistic regression ensures that our estimate for $p(X)$ lies between 0 and 1.



## Maximum Likelihood

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick $\beta_0$ and $\beta_1$ to maximize the likelihood of the observed data.

Most statistical packages can fit linear logistic regression models by maximum likelihood. In `R` we use the `glm` function.

| | Coefficient | Std. Error | Z-statistic | P-value |
|---|---|---|---|---|
| `Intercept` | -10.6513 | 0.3612 | -29.5 | $< 0.0001$ |
| `balance` | 0.0055 | 0.0002 | 24.9 | $< 0.0001$ |

## Making Predictions

What is our estimated probability of `default` for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

With a balance of \$2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$



Lets do it again, using `student` as the predictor.

| | Coefficient | Std. Error | Z-statistic | P-value |
|---|---|---|---|---|
| `Intercept` | -3.5041 | 0.0707 | -49.55 | $< 0.0001$ |
| `student[Yes]` | 0.4049 | 0.1150 | 3.52 | 0.0004 |

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$



## Logistic regression with several variables

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

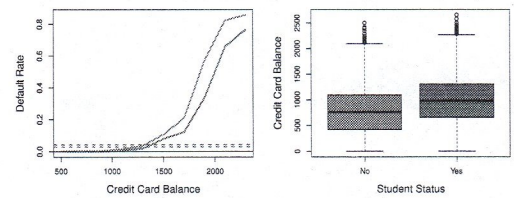
$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

| | Coefficient | Std. Error | Z-statistic | P-value |
|---|---|---|---|---|
| `Intercept` | -10.8690 | 0.4923 | -22.08 | $< 0.0001$ |
| `balance` | 0.0057 | 0.0002 | 24.74 | $< 0.0001$ |
| `income` | 0.0030 | 0.0082 | 0.37 | 0.7115 |
| `student[Yes]` | -0.6468 | 0.2362 | -2.74 | 0.0062 |

Why is coefficient for `student` negative, while it was positive before?



## Confounding



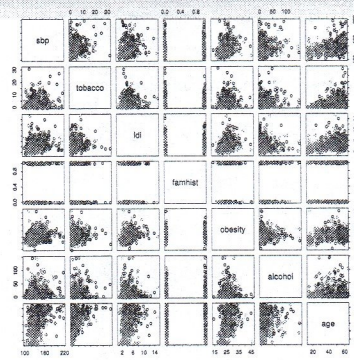

- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
- But for each level of balance, students default less than non-students.
- Multiple logistic regression can tease this out.



## Example: South African Heart Disease

- 160 cases of MI (myocardial infarction) and 302 controls (all male in age range 15-64), from Western Cape, South Africa in early 80s.
- Overall prevalence very high in this region: 5.1%.
- Measurements on seven predictors (risk factors), shown in scatterplot matrix.
- Goal is to identify relative strengths and directions of risk factors.
- This was part of an intervention study aimed at educating the public on healthier diets.






Scatterplot matrix of the *South African Heart Disease* data. The response is color coded — The cases (MI) are red, the controls turquoise. `famhist` is a binary variable, with 1 indicating family history of MI.



```
> heartfit<-glm(chd~.,data=heart,family=binomial)
> summary(heartfit)

Call:
glm(formula = chd ~ ., family = binomial, data = heart)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -4.1295997  0.9641558  -4.283 1.84e-05 ***
sbp             0.0057607  0.0056326   1.023  0.30643
tobacco         0.0795256  0.0262150   3.034  0.00242 **
ldl             0.1847793  0.0574115   3.219  0.00129 **
famhistPresent  0.9391855  0.2248691   4.177 2.96e-05 ***
obesity        -0.0345434  0.0291053  -1.187  0.23529
alcohol         0.0006065  0.0044550   0.136  0.89171
age             0.0425412  0.0101749   4.181 2.90e-05 ***

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 483.17  on 454  degrees of freedom
AIC: 499.17
```



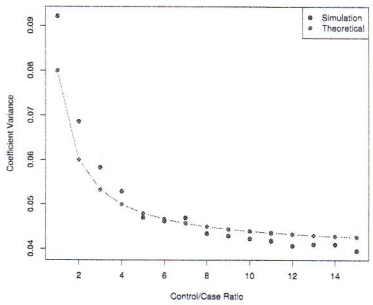
## Case-control sampling and logistic regression

- In South African data, there are 160 cases, 302 controls — $\tilde{\pi} = 0.35$ are cases. Yet the prevalence of MI in this region is $\pi = 0.05$.
- With case-control samples, we can estimate the regression parameters $\beta_j$ accurately (if our model is correct); the constant term $\beta_0$ is incorrect.
- We can correct the estimated intercept by a simple transformation

$$\hat{\beta}_0^* = \hat{\beta}_0 + \log\frac{\pi}{1-\pi} - \log\frac{\tilde{\pi}}{1-\tilde{\pi}}$$

- Often cases are rare and we take them all; up to five times that number of controls is sufficient. See next frame

## Diminishing returns in unbalanced binary data




Sampling more controls than cases reduces the variance of the parameter estimates. But after a ratio of about 5 to 1 the variance reduction flattens out.



## Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package `glmnet`) has the symmetric form

$$\Pr(Y=k|X)=\frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{\ell=1}^{K}e^{\beta_{0\ell}+\beta_{1\ell}X_1+\ldots+\beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.
(The *mathier* students will recognize that some cancellation is possible, and only $K-1$ linear functions are needed as in 2-class logistic regression.)

Multiclass logistic regression is also referred to as *multinomial regression*.



## Discriminant Analysis

Here the approach is to model the distribution of $X$ in each of the classes separately, and then use *Bayes theorem* to flip things around and obtain $\Pr(Y|X)$.

When we use normal (Gaussian) distributions for each class, this leads to linear or quadratic discriminant analysis.

However, this approach is quite general, and other distributions can be used as well. We will focus on normal distributions.



## Bayes theorem for classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probabilistic modeling. Here we focus on a simple result, known as Bayes theorem:

$$\Pr(Y=k|X=x)=\frac{\Pr(X=x|Y=k)\cdot\Pr(Y=k)}{\Pr(X=x)}$$

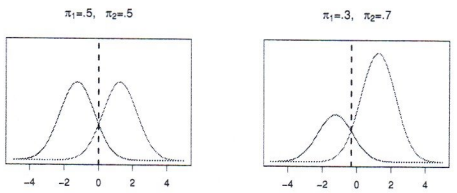
One writes this slightly differently for discriminant analysis:

$$\Pr(Y=k|X=x)=\frac{\pi_k f_k(x)}{\sum_{l=1}^{K}\pi_l f_l(x)},\quad \text{where}$$

- $f_k(x)=\Pr(X=x|Y=k)$ is the *density* for $X$ in class $k$. Here we will use normal densities for these, separately in each class.
- $\pi_k=\Pr(Y=k)$ is the marginal or *prior* probability for class $k$.



## Classify to the highest density




We classify a new point according to which density is highest.

When the priors are different, we take them into account as well, and compare $\pi_k f_k(x)$. On the right, we favor the pink class — the decision boundary has shifted to the left.



## Why discriminant analysis?

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- If $n$ is small and the distribution of the predictors $X$ is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- Linear discriminant analysis is popular when we have more than two response classes, because it also provides low-dimensional views of the data.



## Linear Discriminant Analysis when $p=1$

The Gaussian density has the form

$$f_k(x)=\frac{1}{\sqrt{2\pi}\sigma_k}e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

Here $\mu_k$ is the mean, and $\sigma_k^2$ the variance (in class $k$). We will assume that all the $\sigma_k=\sigma$ are the same.

Plugging this into Bayes formula, we get a rather complex expression for $p_k(x)=\Pr(Y=k|X=x)$:

$$p_k(x)=\frac{\pi_k\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\sum_{l=1}^{K}\pi_l\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2}}$$

Happily, there are simplifications and cancellations.



## Discriminant functions

To classify at the value $X=x$, we need to see which of the $p_k(x)$ is largest. Taking logs, and discarding terms that do not depend on $k$, we see that this is equivalent to assigning $x$ to the class with the largest *discriminant score*:

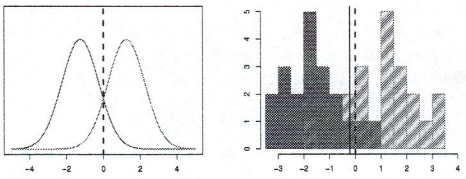
$$\delta_k(x)=x\cdot\frac{\mu_k}{\sigma^2}-\frac{\mu_k^2}{2\sigma^2}+\log(\pi_k)$$

Note that $\delta_k(x)$ is a *linear* function of $x$.

If there are $K=2$ classes and $\pi_1=\pi_2=0.5$, then one can see that the *decision boundary* is at

$$x=\frac{\mu_1+\mu_2}{2}.$$

(See if you can show this)



Example with $\mu_1 = -1.5$, $\mu_2 = 1.5$, $\pi_1 = \pi_2 = 0.5$, and $\sigma^2 = 1$.

Typically we don't know these parameters; we just have the training data. In that case we simply estimate the parameters and plug them into the rule.



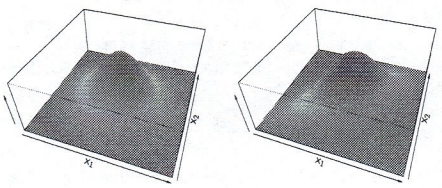
## Estimating the parameters

$$\begin{aligned}\hat{\pi}_k &= \frac{n_k}{n}\\ \hat{\mu}_k &= \frac{1}{n_k}\sum_{i:\, y_i=k} x_i\\ \hat{\sigma}^2 &= \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:\, y_i=k}(x_i-\hat{\mu}_k)^2\\ &= \sum_{k=1}^{K}\frac{n_k-1}{n-K}\cdot\hat{\sigma}_k^2\end{aligned}$$

where $\hat{\sigma}_k^2 = \frac{1}{n_k-1}\sum_{i:\, y_i=k}(x_i-\hat{\mu}_k)^2$ is the usual formula for the estimated variance in the $k$th class.



## Linear Discriminant Analysis when $p > 1$



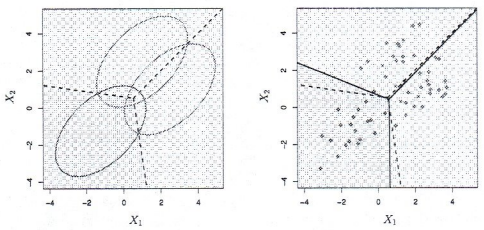
Density: $f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\mathbf{\Sigma}^{-1}(x-\mu)}$

Discriminant function: $\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log\pi_k$

Despite its complex form,
$\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \ldots + c_{kp}x_p$ — a linear function.



## Illustration: $p = 2$ and $K = 3$ classes



Here $\pi_1 = \pi_2 = \pi_3 = 1/3$.
The dashed lines are known as the *Bayes decision boundaries*.
Were they known, they would yield the fewest misclassification errors, among all possible classifiers.

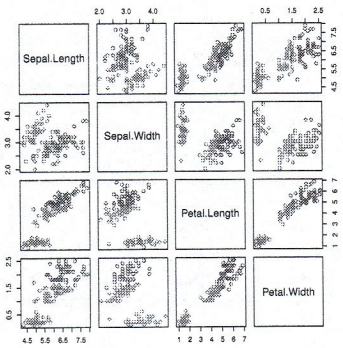


## Fisher's Iris Data

4 variables
3 species
50 samples/class

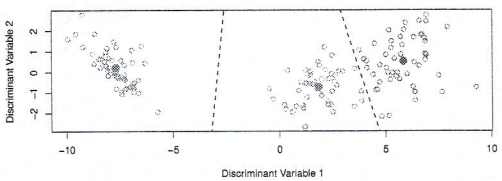
- Setosa
- Versicolor
- Virginica

LDA classifies all but 3 of the 150 training samples correctly.






## Fisher's Discriminant Plot




When there are $K$ classes, linear discriminant analysis can be viewed exactly in a $K-1$ dimensional plot.
Why? Because it essentially classifies to the closest centroid, and they span a $K-1$ dimensional plane.
Even when $K > 3$, we can find the "best" 2-dimensional plane for visualizing the discriminant rule.



## From $\delta_k(x)$ to probabilities

Once we have estimates $\hat{\delta}_k(x)$, we can turn these into estimates for class probabilities:

$$\widehat{\Pr}(Y = k|X = x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_{l=1}^{K} e^{\hat{\delta}_l(x)}}.$$

So classifying to the largest $\hat{\delta}_k(x)$ amounts to classifying to the class for which $\widehat{\Pr}(Y = k|X = x)$ is largest.

When $K = 2$, we classify to class 2 if $\widehat{\Pr}(Y = 2|X = x) \geq 0.5$, else to class 1.

## LDA on Credit Data

| | | True Default Status: No | Yes | Total |
|---|---|---|---|---|
| *Predicted Default Status* | No | 9644 | 252 | 9896 |
| | Yes | 23 | 81 | 104 |
| | Total | 9667 | 333 | 10000 |

Confusion matrix

$(23 + 252)/10000$ errors — a 2.75% misclassification rate!

Some caveats:

- This is *training* error, and we may be overfitting. Not a big concern here since $n = 10000$ and $p = 2$!
- If we classified to the prior — always to class No in this case — we would make 333/10000 errors, or only 3.33%.
- Of the true No's, we make $23/9667 = 0.2\%$ errors; of the true Yes's, we make $252/333 = 75.7\%$ errors!

what matters are the costs

we lose MONEY

we lose CLIENTS → different costs



Is this good?
We have 333/1000 of "Yes";

we estimate priors from data assuming that it's a good sample

$\mathbb{P}(L = No) = 9667/10000$
$\mathbb{P}(L = Yes) = 333/10000 = 0.03$ ⇒ error rate of flipping the coin = 0.03

we have to compare 2.75% with this error rate!

⇒ a coin that gives "Yes" only 3% of the times is equal to this method that gives a misclassification rate of 2.75%

## Types of errors

False positive rate: The fraction of negative examples that are classified as positive — 0.2% in example.

False negative rate: The fraction of positive examples that are classified as negative — 75.7% in example.

We produced this table by classifying to class `Yes` if

$$\widehat{\Pr}(\texttt{Default} = \texttt{Yes}|\texttt{Balance}, \texttt{Student}) \geq 0.5$$

We can change the two error rates by changing the threshold from 0.5 to some other value in $[0, 1]$:

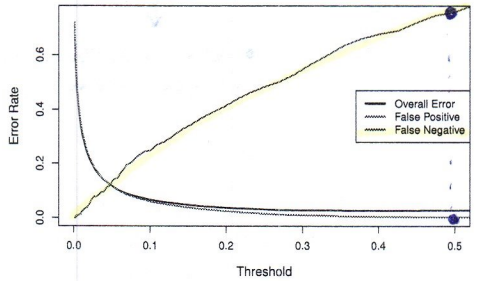
$$\widehat{\Pr}(\texttt{Default} = \texttt{Yes}|\texttt{Balance}, \texttt{Student}) \geq \textit{threshold},$$

and vary *threshold*.



this is equivalent to changing the costs of misclassification

## Varying the *threshold*

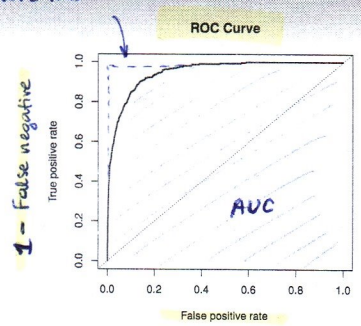



by changing threshold we change false positive and negative

In order to reduce the false negative rate, we may want to reduce the threshold to 0.1 or less.



we would love this




for every threshold we have a point on the curve

The *ROC plot* displays both simultaneously.

Sometimes we use the *AUC* or *area under the curve* to summarize the overall performance. Higher *AUC* is good.



## Other forms of Discriminant Analysis

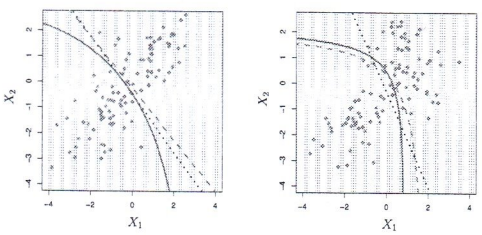
$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

When $f_k(x)$ are Gaussian densities, with the same covariance matrix $\mathbf{\Sigma}$ in each class, this leads to linear discriminant analysis. By altering the forms for $f_k(x)$, we get different classifiers.

- With Gaussians but different $\mathbf{\Sigma}_k$ in each class, we get *quadratic discriminant analysis*.
- With $f_k(x) = \prod_{j=1}^{p} f_{jk}(x_j)$ (conditional independence model) in each class we get *naive Bayes*. For Gaussian this means the $\mathbf{\Sigma}_k$ are diagonal.
- Many other forms, by proposing specific density models for $f_k(x)$, including nonparametric approaches.



## Quadratic Discriminant Analysis




$$\delta_k(x) = -\frac{1}{2}(x - \mu_k)^T \mathbf{\Sigma}_k^{-1}(x - \mu_k) + \log \pi_k - \frac{1}{2} \log |\Sigma_k|$$

Because the $\mathbf{\Sigma}_k$ are different, the quadratic terms matter.



## Naive Bayes

Assumes features are independent in each class.

Useful when $p$ is large, and so multivariate methods like QDA and even LDA break down.

- Gaussian naive Bayes assumes each $\mathbf{\Sigma}_k$ is diagonal:

$$\begin{aligned} \delta_k(x) &\propto \log\left[\pi_k \prod_{j=1}^{p} f_{kj}(x_j)\right] \\ &= -\frac{1}{2}\sum_{j=1}^{p}\left[\frac{(x_j - \mu_{kj})^2}{\sigma_{kj}^2} + \log \sigma_{kj}^2\right] + \log \pi_k \end{aligned}$$

- can use for *mixed* feature vectors (qualitative and quantitative). If $X_j$ is qualitative, replace $f_{kj}(x_j)$ with probability mass function (histogram) over discrete categories.

Despite strong assumptions, naive Bayes often produces good classification results.



## Logistic Regression versus LDA

For a two-class problem, one can show that for LDA

$$\log\left(\frac{p_1(x)}{1 - p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1 x_1 + \ldots + c_p x_p$$

So it has the same form as logistic regression.

The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on $\Pr(Y|X)$ (known as *discriminative learning*).
- LDA uses the full likelihood based on $\Pr(X, Y)$ (known as *generative learning*).
- Despite these differences, in practice the results are often very similar.

Footnote: logistic regression can also fit quadratic boundaries like QDA, by explicitly including quadratic terms in the model.



## Summary

- Logistic regression is very popular for classification, especially when $K = 2$.
- LDA is useful when $n$ is small, or the classes are well separated, and Gaussian assumptions are reasonable. Also when $K > 2$.
- Naive Bayes is useful when $p$ is very large.
- See Section 4.5 for some comparisons of logistic regression, LDA and KNN.

Logistic regression has a problem when there is a neat separation between the groups (it's not able to find the parameters), while LDA is very good with a neat separation.

**Logistic regression** vs. **Classification**

Tryies to estimate directly the posterior distribution without going through this modelling procedure

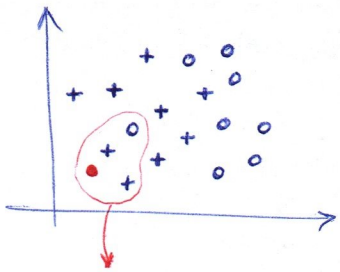
- priors probabilities
- distribution of the features in each group

↓

- posterior distribution (Bayes theorem)

## KNN

suppose $k=3$ ⇒ we pick the 3 closest elements



if we consider a new element •,
we look for the k nearest neighbors
and we do an average
In this case • is +

Problem (frequent problem) of KNN: overfitting

How do we chose k? By cross-validation (≠ k-folder cross validation)