Modelos Estadísticos para la Regresión y la Clasificación

Clase 8: Clasificación: Analisis Discriminante, Regresión Logística y kNN

Mathias Bourel

Instituto de Matemática y Estadística Prof. Rafael Laguardia (IMERL)
Facultad de Ingeniería, Universidad de la República, Uruguay

7 de septiembre de 2024

Introducción

$\mathcal{L} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ muestra de datos.

- **Discriminar:** usar $\mathcal{L}$ para construir un clasificador (función de las caracteristicas X_i) para separar lo mejor posibles los grupos dados.
- **Clasificar:** usar el clasificador para predecir la etiqueta Y_{new} de una nueva observación X_{new}.

Suponemos que hay dos grupos G_1 y G_2 y que cada individuo pertenece a un único grupo (por ejemplo sano/enfermo, spam/no spam).

Plan

Análisis Discriminante Lineal Gaussiano (LDA)

Supongamos que $f_1 \sim N(\mu_1, \Sigma)$ y $f_2 \sim N(\mu_2, \Sigma)$ - misma matriz de covarianzas-. De

$$\frac{f_2(x)\pi_2}{c(2|1)} > \frac{f_1(x)\pi_1}{c(1|2)}$$

tomando logaritmo tenemos que

$$-\frac{1}{2}\underbrace{(x-\mu_2)'\Sigma^{-1}(x-\mu_2)}_{D_2^2} + \log\left(\frac{\pi_2}{c(2|1)}\right) > -\frac{1}{2}\underbrace{(x-\mu_1)'\Sigma^{-1}(x-\mu_1)}_{D_1^2} + \log\left(\frac{\pi_1}{c(1|2)}\right) \quad (*)$$

donde D_i^2 es la distancia de Mahalanobis entre el punto observado x y la media de la población i (recordar los slides sobre normal multivariada). Entonces:

$$D_1^2 - 2\log\left(\frac{\pi_1}{c(1|2)}\right) > D_2^2 - 2\log\left(\frac{\pi_2}{c(2|1)}\right)$$

y si suponemos que $\pi_1 = \pi_2$ y los costes iguales, clasificamos en la población 2 si

$$D_1^2 > D_2^2$$

Obs: si $\Sigma = \sigma^2 I$ entonces la regla equivale en usar la distancia euclidea.

Volviendo a $(*)$, si desarrollamos, al tener la misma matriz de varianzas-covarianzas Σ, se elimina el termino cuadrático $x'\Sigma^{-1}x$. Entonces

$$-\mu_1'\Sigma^{-1}x + \frac{1}{2}\mu_1'\Sigma^{-1}\mu_1 > -\mu_2'\Sigma^{-1}x + \frac{1}{2}\mu_2'\Sigma^{-1}\mu_2 - \log\left(\frac{c(1|2)\pi_2}{c(2|1)\pi_1}\right)$$

$$(\mu_2 - \mu_1)'\Sigma^{-1}x > (\mu_2 - \mu_1)'\Sigma^{-1}\left(\frac{\mu_1 + \mu_2}{2}\right) - \log\left(\frac{c(1|2)\pi_2}{c(2|1)\pi_1}\right)$$

Si $w = \Sigma^{-1}(\mu_2 - \mu_1)$, entonces

$$w'x > \underbrace{w'\left(\frac{\mu_1 + \mu_2}{2}\right) - \log\left(\frac{c(1|2)\pi_2}{c(2|1)\pi_1}\right)}_{-w_0}$$

clasificamos en la población 2 si

$$w'x > -w_0 \Rightarrow L(x) = w'x + w_0 > 0$$

Suponiendo costes y probabilidades a priori iguales, volviendo a $w'x > \underbrace{w'\left(\frac{\mu_1+\mu_2}{2}\right)}_{-w_0}$ (es decir $L(x) > 0$) entonces:

$$w'x - w'\mu_1 > w'\mu_2 - w'x$$

Entonces el procedimiento para clasificar el individuo x_0 en P_1 o en P_2 según este método es el siguiente:

1. Calcular el vector $w = \Sigma^{-1}(\mu_2 - \mu_1)$.
2. Construir la variables indicadora discriminante $z = w'x$
3. Clasificar en la población donde la distancia $|z_0 - m_i|$ es mínima siendo $z_0 = w'x_0$ y $m_i = w'\mu_i$.

Observar que:

- $Var(z) = Var(w'x) = w'Var(x)w = w'\Sigma w = \underbrace{(\mu_2 - \mu_1)'\Sigma^{-1}(\mu_2 - \mu_1)}_{D^2}$
- Por otro lado:
  $m_2 - m_1 = w'(\mu_2 - \mu_1) = (\Sigma^{-1}(\mu_2 - \mu_1))'(\mu_2 - \mu_1) = (\mu_2 - \mu_1)'\Sigma^{-1}(\mu_2 - \mu_1) = D^2$

Entonces

$$Var(z) = m_2 - m_1$$

Podemos interpretar a la variable z de la siguiente manera:
si dividimos la relación $w'x - w'\mu_1 > w'\mu_2 - w'x$ por $||w||$ y $u = \frac{w}{||w||}$ entonces

$$u'x - u'\mu_1 > u'\mu_2 - u'x$$

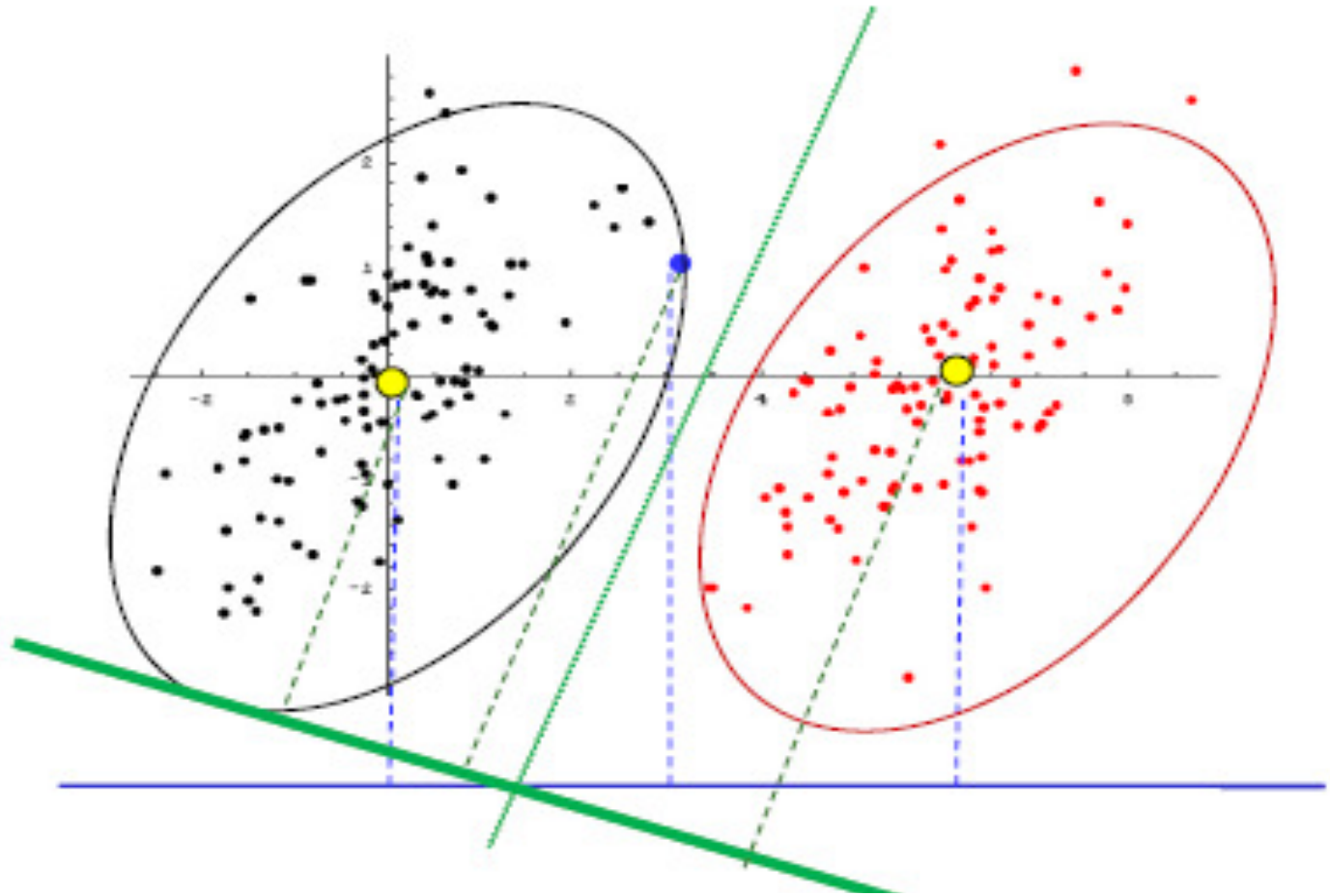
y $\widehat{P}_u(x) = u'x$ es la proyección (el escalar) de x en la dirección u, y $u'\mu_i$ es la proyección de μ_i en la dirección u para $i = 1, 2$. Entonces elegiremos la población 2 si

$$\widehat{P}_u(x) > \widehat{P}_u\left(\frac{\mu_1 + \mu_2}{2}\right)$$

(el hiperplano perpendicular a u por $u'\left(\frac{\mu_1+\mu_2}{2}\right)$ divide el espacio muestral en dos regiones)

Interpretación geométrica del Análisis Discriminante Lineal

En la figura siguiente representamos la situación establecida en la transparencia anterior: proyectando el punto medio de las medias sobre u (el punto medio de los dos puntos amarillos), y proyectando x (el punto azul) sobre u sabremos cuál de las dos poblaciones atribuirle.



Cálculo de probabilidades de error

Recordamos que la variable $z = w'x$ tiene esperanza $\mathbb{E}(z) = m_i = w'\mu_i$ y varianza $D^2 = m_2 - m_1$. Entonces

$$\mathbb{P}(2|1) = \mathbb{P}\left(z \geq \frac{m_1 + m_2}{2} | z \sim N(m_1, D)\right) = \mathbb{P}\left(y \geq \frac{\frac{m_1+m_2}{2} - m_1}{D} | y \sim N(0,1)\right)$$

$$\mathbb{P}(2|1) = 1 - \Phi\left(\frac{D}{2}\right)$$

$$\mathbb{P}(1|2) = \mathbb{P}\left(z \leq \frac{m_1 + m_2}{2} | z \sim N(m_2, D)\right) = \mathbb{P}\left(y \leq \frac{\frac{m_1+m_2}{2} - m_2}{D} | y \sim N(0,1)\right)$$

$$\mathbb{P}(1|2) = \Phi\left(-\frac{D}{2}\right)$$

Las probabilidades de error son iguales, el error de clasificación sólo depende de la distancia de Mahalanobis entre las medias.

Probabilidads a posteriori

Volviendo a la cuenta del principio:

$$\mathbb{P}(1|x) = \frac{\pi_1 f_1(x)}{\pi_1 f_1(x) + \pi_2 f_2(x)}$$

$$\mathbb{P}(1|x) = \frac{\pi_1 \exp\left(-\frac{1}{2}(x-\mu_1)'\Sigma^{-1}(x-\mu_1)\right)}{\pi_1 \exp\left(-\frac{1}{2}(x-\mu_1)'\Sigma^{-1}(x-\mu_1)\right) + \pi_2 \exp\left(-\frac{1}{2}(x-\mu_2)'\Sigma^{-1}(x-\mu_2)\right)}$$

$$\mathbb{P}(1|x) = \frac{1}{1 + \frac{\pi_2}{\pi_1} \exp\left(-\frac{1}{2}(D_2^2 - D_1^2)\right)}$$

En el caso que las probabilidades a priori sean iguales, cuanto más alejado está el punto de la población 1, $(D_1^2 > D_2^2)$, el denominador es más grande y menor será $\mathbb{P}(1|x)$ y al contrario.

Ejemplo (Peña, pág 406)

Retrato entre dos posibles pintores. Se miden dos variables: X_1 profundidad del trazo y X_2 proporción que ocupa el retrato sobre la superficie del lienzo.

Retratos del pintor A $\sim N\left(\mu_A = \begin{pmatrix} 2 \\ 0{,}8 \end{pmatrix}, \Sigma\right)$, Restratos de pintor B $\sim N\left(\mu_B = \begin{pmatrix} 2{,}3 \\ 0{,}7 \end{pmatrix}, \Sigma\right)$

Covarianzas $\Sigma = \begin{pmatrix} 0{,}25 & 0{,}025 \\ 0{,}025 & 0{,}01 \end{pmatrix}$, nueva obra a clasificar $x_0 = \begin{pmatrix} 2{,}1 \\ 0{,}75 \end{pmatrix}$

$$D_A^2 = \begin{pmatrix} 2{,}1-2 & 0{,}75-0{,}8 \end{pmatrix} \begin{pmatrix} 0{,}25 & 0{,}025 \\ 0{,}025 & 0{,}01 \end{pmatrix}^{-1} \begin{pmatrix} 2{,}1-2 \\ 0{,}75-0{,}8 \end{pmatrix} = 0{,}52$$

$$D_B^2 = \begin{pmatrix} 2{,}1-2{,}3 & 0{,}75-0{,}7 \end{pmatrix} \begin{pmatrix} 0{,}25 & 0{,}025 \\ 0{,}025 & 0{,}01 \end{pmatrix}^{-1} \begin{pmatrix} 2{,}1-2{,}3 \\ 0{,}75-0{,}7 \end{pmatrix} = 0{,}8133$$

Entonces

$$\mathbb{P}(A|x) = \frac{1}{1+\exp\left(-\frac{1}{2}(D_B^2 - D_A^2)\right)} = \frac{1}{1+\exp\left(-\frac{1}{2}(0,\,8133 - 0{,}52)\right)} = 0,\,5376$$

$$\mathbb{P}(A|B) = 1 - \Phi\left(\frac{D^2}{2}\right) = 1 - \Phi\left(\frac{\begin{pmatrix} 2-2{,}3 & 0{,}8-0{,}7 \end{pmatrix}\begin{pmatrix} 0{,}25 & 0{,}025 \\ 0{,}025 & 0{,}01 \end{pmatrix}^{-1}\begin{pmatrix} 2-2{,}3 \\ 0{,}8-0{,}7 \end{pmatrix}}{2}\right) = 1 - \Phi(0{,}808) = 0,\,189$$

LDA en R

```
Iris <- data.frame(rbind(iris3[,,1], iris3[,,2], iris3[,,3]),
                   Sp = rep(c("s","c","v"), rep(50,3)))
train <- sample(1:150, 75)
table(Iris$Sp[train])
## your answer may differ
##  c  s  v
## 22 23 30
z <- lda(Sp ~ ., Iris, prior = c(1,1,1)/3, subset = train)
predict(z, Iris[-train, ])$class
##  [1] s s s s s s s s s s s s s s s s s s s s s s s s s c c c
## [31] c c c c c c c v c c c c v c c c c c c c c c c c v v v v v
## [61] v v v v v v v v v v v v v v v
```

Análisis Discriminate Cuadrático (QDA)

Si suponemos que las matrices de covarianzas no son iguales, el clasificador de Bayes asigna la observación x a la clase para la cual:

$$-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)+\log(\pi_k)=-\frac{1}{2}x'\Sigma_k^{-1}x+x'\Sigma_k^{-1}\mu_k-\frac{1}{2}\mu_k'\Sigma_k^{-1}\mu_k+\log(\pi_k)$$

es mayor (observe que esto es una función cuadrática).

Plan

Introducción

Supongamos que queremos modelar una variable Y categórica, binaria, ($Y \sim B(1,p)$) por ejemplo:

- Presencia/ausencia de una determinada especie
- Especie1/especie2
- Enfermo/ no enfermo
- Spam/no spam
- Quiebra/no-quiebra

Si quisieramos usar la regresión lineal

$$Y = \beta_0 + \beta_1 X + \epsilon$$

$$\mathbb{E}(Y|X = x_i) = \beta_0 + \beta_1 x_i$$

$$\mathbb{E}(Y|X = x_i) = \underbrace{\mathbb{P}(Y = 1|X = x_i)}_{p_i} \times 1 + \mathbb{P}(Y = 0|X = x_i) \times 0 = p_i$$

y por lo tanto

$$p_i = \beta_0 + \beta_1 x_i$$

O sea la predicción hecha por el modelo estima la probabilidad de que el individuo x_i pertenezca a la población 1. Inconveniente: $p_i \in [0,1]$....No parece apropiado...

Definición de los Modelos Lineales Generalizados

Volviendo al caso general, si llamamos $\mu = \mathbb{E}(Y|X)$, y consideramos una función g monótona y diferenciable entonces

$$g(\mu) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = \nu$$

Un modelo lineal generalizado está dado por

$$g(\mu) = X'\beta = \nu$$

donde ν es un predictor lineal. Todo GLM tiene:

- una componente aleatoria: variable de respuesta Y, representada por μ.
- una componente sistemática: combinación lineal de las variables explicativas (independientes, predictoras).
- Función link o de enlace: relaciona las dos componentes anteriores.

Ejemplo, regresión logística

Ejemplo: modelar la probabilidad de fraude por impago (default) en función del balance de la cuenta bancaria (balance), el ingreso (income) y si es estudiante o no.

```
>library(ISLR)
>attach(Default)
>data=Default
>data
>head(data,n=4)
 default student   balance   income
1      No      No  729.5265 44361.63
2      No     Yes  817.1804 12106.13
3      No      No 1073.5492 31767.14
4      No      No  529.2506 35704.49
```

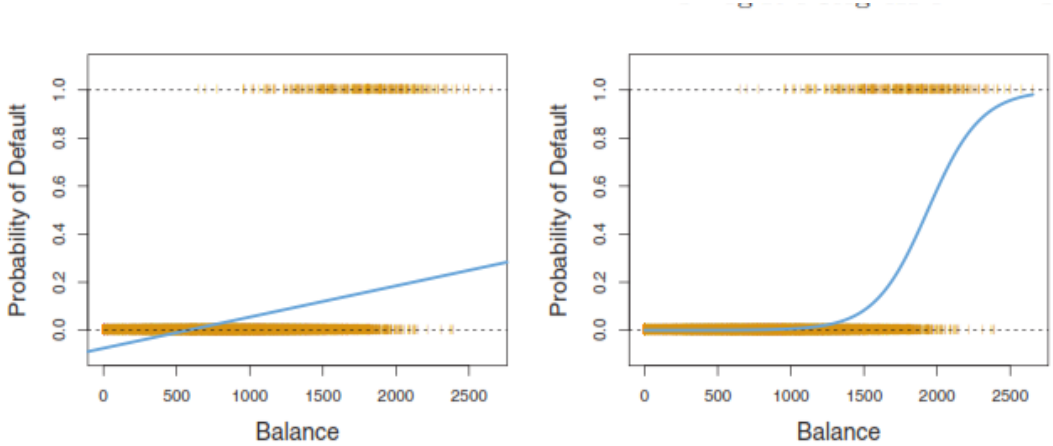


Figura: Ejemplo: Probabilidad de "default" en la tarjeta de crédito en función del balance mensual en la tarjeta (Cap. 4 de [2])

Claramente la recta de regresión lineal no se ajusta bien a los datos por lo cual preferimos una sigmoide.

Modelo Logístico

Consideramos al modelo de regresión logística binaria. Vamos a querer que

$$p(x_i) = p_i = F(\beta_0 + \beta_1 x_i)$$

donde F es una función de distribución para que el modelo proporcione directamente la probabilidad de pertenecer a cada uno de los grupos.
Una función adecuada para modelar este tipo de variables es la función *logit* quedando el modelo de regresión logística con la siguiente forma: si $p = \mathbb{P}(Y = 1|X)$ entonces el modelo logístico múltiple es:

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

Entonces

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

En este caso $F(t) = \frac{1}{1+e^{-t}}$ y se llama función de distribución logística (función sigmoide).

Estimación de los parámetros en regresión logística

A partir de n observaciones y suponiendo que

$$g(\mu_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} \quad i = 1, \ldots, n$$

se buscan los paramétros $\theta = (\beta_0, \beta_1, \ldots, \beta_d)$ que maximicen el logaritmo de la función de verosimilitud L:

$$\ln(L(y, \theta)) = \ln\left(\prod_{i=1}^{n} f(y_i, \theta)\right) = \sum_{i=1}^{n} \ln f(y_i, \theta)$$

En el caso de la regresión lógistica binaria, y suponiendo que el modelo es $\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \mathbf{x}$, la función de verosimilitud se calcula como:

$$L(y, \theta) = L(y, \beta_0, \beta_1) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1-y_i}$$

y

$$\ln(L(y, \beta_0, \beta_1)) = \sum_{i=1}^{n} (y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i))$$

Acordarse que $p_i = \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)} \quad \forall\, i = 1, \ldots, n.$

Estimación de los parámetros

Se prueba que se encuentra un único vector β que anula a todas las derivadas parciales de $L(y, \theta)$ simultaneamente. Ese β resulta ser un óptimo del problema de maximización.

$$(\widehat{\beta}_0, \widehat{\beta}_1) = \underset{\beta_0, \beta_1}{\text{Argmax}} \ln(L(y, \beta_0, \beta_1))$$

En la práctica este estimador se calcula usando métodos iterativos (método de Newton-Raphson o Fisher scoring) o método de descenso por gradiente.

Estimación e interpretación de los coeficientes

- La estimación de los coeficientes de la regresión hecha por el método de máxima verosimilitud se aplica también para cualquier GLM. Idem la comparación de modelos.
- Supongamos que $\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \mathbf{x}$ donde $p = p(\mathbf{x}) = \mathbb{P}(Y = 1|X = \mathbf{x})$
  - $\text{Odds}(\mathbf{x}) = \frac{p(\mathbf{x})}{1-p(\mathbf{x})}$ indica cuantas veces es más probable que ocurra $Y = 1$ respecto a que no ocurra (o sea ocurra $Y = 0$).
  - Odds-ratio: $\text{OR}(\mathbf{x}) = \frac{Odds(\mathbf{x}+1)}{Odds(\mathbf{x})}$ (como cambia la respuesta de interés al aumentar una unidad). En este caso una cuenta inmediata muestra que $\beta_1 = \ln(OR)$ y por lo tanto
  $$e^{\beta_1} = OR$$
  Entonces si X augmenta de k unidades se tiene que $OR = e^{k\beta_1}$.
  - Odds-ratio$(\mathbf{x_i}, \mathbf{x_j}) = \frac{Odds(\mathbf{x_i})}{Odds(\mathbf{x_j})} = e^{\beta_1(\mathbf{x_j} - \mathbf{x_i})}$ y en general es igual a $e^{\beta'(\mathbf{x_j} - \mathbf{x_i})}$

  Por ejemplo si:
  - la probabilidad de tener cancer de pulmon para un fumador es $\mathbb{P}(Y = 1|X = \text{fumador}) = 0{,}01$ por lo que $\mathbb{P}(Y = 0|X = \text{fumador}) = 0{,}99$ y odds(X =fumador)=1/99.
  - la probabilidad de tener cancer de pulmon para un no fumador es $\mathbb{P}(Y = 1|X = \text{no fumador}) = 10^{-4}$
  - OR(fumador, no fumador)= $\frac{1/99}{1/9999} = 101$ y hay 101 veces más chance de tener cancer de púlmon para un fumador que para un no fumador.

Estimación e interpretación de los coeficientes

- Recordemos que en la regresión lineal el coeficiente β_j asociado a una determinada variable X_j indicaba el cambio en la variable Y al aumentar una unidad de la variable X_j manteniendo las demás fijas.
- Aquí, lo que nos dice este coeficiente es el cambio en el log(p/1-p) al aumentar en una unidad la X_j.
- Una forma de interpretar los coeficientes β_j de forma genérica es: si son positivos, entonces al aumentar X_j aumenta la probabilidad de ocurrencia de default, si son negativos, al aumentar X_j, esta probabilidad disminuye.

Test sobre significancia modelo

Se puede usar la desvianza nula y la desvianza residual para testear la significancia del modelo:

$$(H_0): \quad \beta_j = 0 \, \forall \, j = 1, \ldots, p$$
$$(H_1): \quad \exists \beta_j \neq 0$$

Bajo la hipótesis nula (H_0) el logaritmo del cociente de las verosimilitudes $-2 \ln \left(\frac{L_{\text{nulo}}}{L_{\text{completo}}} \right) \sim \chi^2_p$

$-2 \ln \left(\frac{L_{\text{nulo}}}{L_{\text{completo}}} \right) = -2 \ln(L_{\text{nulo}}) + 2 \ln(L_{\text{completo}}) = \text{Null deviance} - \text{Residual deviance}$

```
>modelo.null=glm(default~1,data=Default, family='binomial')
>modelo3=glm(default~.,data=Default, family='binomial')
> anova(modelo.null,modelo3,test='Chisq')
Analysis of Deviance Table

Model 1: default ~ 1
Model 2: default ~ student + balance + income
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      9999     2920.7
2      9996     1571.5  3   1349.1 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como el p-valor es menor que 0,05 hay suficiente evidencia para rechazar la hipotesis nula
Teniendo en cuenta que los grados de libertad son $p = 3$, otra posibilidad es hacer:

```
> chi= modelo.null$deviance - modelo3$deviance
> pchisq(chi, df=3,lower.tail=F)
```

Modelos Anidados

Esto se puede extender a la comparación de dos modelos anidados

$$\begin{cases} (H_0): & \beta = (\beta_0, \beta_1, \ldots, \beta_q) \\ (H_1): & \beta = (\beta_0, \beta_1, \ldots, \beta_p) \end{cases}$$

con $q < p < n$. La idea es que si la hipótesis nula es cierta entonces las verosimilitudes deben ser muy cercanos en valor y por lo tanto la diferencia entre los logaritmos chica. Usamos las diferencias entre las desvianzas

$$D_0 - D_1 = 2\left(\ln(L(\widehat{\beta}_p, y)) - \ln(L(\widehat{\beta}_q, y))\right)$$

Si los dos modelos describen bien los datos entonces $D_0 \sim \chi^2_{n-(q+1)}$ y $D_1 \sim \chi^2_{n-(p+1)}$ por lo tanto $D_0 - D_1 \sim \chi^2_{p-q}$ y rechazamos la hipotesis nula si $D_0 - D_1 > \chi^2_{p-q}$.

Modelos Anidados

Aca vemos la prueba chi-cuadrado a medida que vamos añadiendo las variables.

```
> anova(modelo3,test='Chisq')
Analysis of Deviance Table

Model: binomial, link: logit

Response: default

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                    9999     2920.7
student  1    11.97      9998     2908.7 0.0005416 ***
balance  1  1337.00      9997     1571.7 < 2.2e-16 ***
income   1     0.14      9996     1571.5 0.7115139
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Criterio Akaike

Al igual que vimos para los modelos lineales (ML), podemos usar el criterio de Akaike (AIC) para comparar modelos con distinto número de parámetros.

$$AIC = -2\ln(L(y, \widehat{\beta}_p)) + 2(p+1)$$

Recordemos que cuanto menor el valor del AIC, mejor es el ajuste.
El número AIC por sí solo no nos dice nada, lo que nos interesa es la diferencia de AIC entre diferentes modelos.

Criterio posible:
diferencias de 0 a 2: modelos similares
diferencias de 4 a 7: es mejor el modelo con menor AIC
diferencias $>$ 10: es mucho mejor el modelo con menor AIC

Si tenemos muchas variables podemos usar selección de variables con los métodos stepwise (paso a paso), forward (hacia adelante) o backward (hacia atrás) tomando como criterio de selección en cada paso el valor del AIC.

Comparación de modelos

```
> anova(modelo,modelo3,test='Chisq')
Analysis of Deviance Table

Model 1: default ~ balance
Model 2: default ~ student + balance + income
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      9998     1596.5
2      9996     1571.5  2   24.907 3.904e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> AIC(modelo)-AIC(modelo3)
[1] 20.90686
```

Hay evidencia por el test chi-cuadrado que el modelo 3 es mejor que el modelo1 y por otro lado también con el criterio del AIC

Seguimos con el ejemplo

```
> modelo=glm(default~balance,family=binomial,data)
> summary(modelo)
Call:
glm(formula = default ~ balance, family = binomial, data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2697  -0.1465  -0.0589  -0.0221   3.7589

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
balance      5.499e-03  2.204e-04   24.95   <2e-16 ***
```

El test de hipótesis que aparece es el Test de Wald con estadístico $\frac{(\widehat{\beta}_j - b_j)^2}{\text{Var}(\widehat{\beta}_j)} \to \chi^2(1)$ bajo $(H_0) : \beta_j = 0$. En lo que nos proporciona R, tenemos $\frac{\widehat{\beta}_j}{\text{s.e}(\widehat{\beta}_j)} \to \mathcal{N}(0,1)$

- $\widehat{\beta}_1 = 0{,}0055 \Rightarrow$ incremento en balance implica incremento en default.
- El estadistico $z = \widehat{\beta}_1 / s.e(\widehat{\beta}_1)$ juega el mismo papel que el estadístico t de la regresión lineal por lo que un valor importante de z implica rechazar la hipotesis nula $(H_0) : \beta_1 = 0$. Esta hipótesis nula implica que $p(X) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$ y no depende del valor de X (en este caso balance).
- Claramente hay una relación entre balance y default.
- Predicción:

```
predict(modelo, data.frame(balance=c(1000,2000)), type='response')
0.005752145 0.585769370
```

  Esto es que $\widehat{p}(1000) = \frac{e^{-10{,}63+0{,}0055\times 1000}}{1+e^{-10{,}63+0{,}0055\times 1000}} = 0{,}00575, \qquad \widehat{p}(2000) = 0{,}586$

- Ajuste:

  `>1-pchisq(modelo$deviance,9998)`

  Obtenemos el valor 1 entonces el modelo se ajusta bien a los datos.
- Si queremos calcular el riesgo de default al aumentar el balance en 100 dólares se tiene que $e^{100\times 0{,}0055} = 1{,}73$ y el riesgo aumenta aproximadamente 2 veces.

Usando la variable categorica "student"

```
>modelo2=glm(default~student,family=binomial,data)
>summary(modelo2)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
studentYes   0.40489    0.11502    3.52 0.000431 ***


>predict(modelo2, data.frame(student=c("Yes","No")), type='response')
0.04313859 0.02919501
```

$$\mathbb{P}(\text{default=Yes}|\text{student=Yes}) = \frac{e^{-3,5041+0,4049\times 1}}{1+e^{-3,5041+0,4049\times 1}} = 0,0431$$

$$\mathbb{P}(\text{default=Yes}|\text{student=No}) = \frac{e^{-3,5041+0,4049\times 0}}{1+e^{-3,5041+0,4049\times 0}} = 0,0292$$

Usando todas las variables:

```
> modelo3=glm(default~.,family=binomial,data)
> summary(modelo3)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
```

```
>predict(modelo3, data.frame( student="Yes",balance=1500,income=40000), type='response')

0.05788194
```

$$\widehat{p}(X) = \frac{e^{-10,869+0,00574\times 1500+0,003\times 40-0,6468\times 1}}{1+e^{-10,869+0,00574\times 1500+0,003\times 40-0,6468\times 1}} = 0,058$$

Si Student=No: $\widehat{p}(X) = \frac{e^{-10,869+0,00574\times 1500+0,003\times 40-0,6468\times 0}}{1+e^{-10,869+0,00574\times 1500+0,003\times 40-0,6468\times 0}} = 0,105$

Las predicciones se hacen generalmente con la regla del *máximo a posteriori*, es decir predecimos el valor de Y por la modalidad k que maximiza la probabilidad $\mathbb{P}(Y = k|X = \mathbf{x})$.
En presencia de dos clases, podríamos pensar que si la probabilidad de estar en una clase es mayor que $1/2$, entonces esa debe ser la clase asignada a $\mathbf{x}$. Pero esta elección de $1/2$ es totalmente arbitraria. Se podría definir la regla de asignación con umbral s como

$$y_s^* = \begin{cases} 1 & \text{si } \mathbb{P}(Y = 1|X = \mathbf{x}) \geq s \\ 0 & \text{si no} \end{cases}$$

¿Qué tan performante es el modelo? ¿Cuánto se equivoca?

```
>pred=predict(modelo3,data.frame=data,type="response")
>contrasts(default)
>prediccion=rep("No",10000)
>prediccion[pred>.5]="Yes"
>table(prediccion,default)
```

```
> table(prediccion,default)
          default
prediccion   No  Yes
       No  9627  228
       Yes   40  105
> mean(prediccion==default)
[1] 0.9732
```

Es satisfactorio??....

Tabla comparativa LM y GLM

	LM	GLM
Parámetros	$\beta_0, \beta_1, \ldots, \beta_p$	$\beta_0, \beta_1, \ldots, \beta_p$
Estimación	Mínimos Cuadrados	Máxima Verosimilitud
Ajuste	R^2	Desvianza
Comparación modelos	AIC, F	AIC, Desvianza
Supuestos	Residuos normales +Gauss	Y familia exponencial

Modelo logístico multiclasses

Supongamos ahora que Y tiene K modalidades con $K > 2$. Sea $\pi_k = \mathbb{P}(Y = k|X = \mathbf{x})$. Nos fijamos una modalidad de referencia, generalmente la última, K, y hacemos $K - 1$ regresiones logísticas de $\pi_k(\mathbf{x})$ vs $\pi_K(\mathbf{x})$:

$$\ln\left(\frac{\pi_k(\mathbf{x})}{\pi_K(\mathbf{x})}\right) = \beta_{0k} + \sum_{j=1}^{p} \beta_{jk} x_j \quad \forall\, 1 \leq k \leq K-1$$

La clasificación se hace asignando a $\mathbf{x}$ la clase con la máxima probabilidad a posteriori, es decir, calculamos las K probabilidades a posteriori siguiente:

$$\mathbb{P}(Y = k|X = \mathbf{x}) = P(Y = K|X = x)e^{\beta_{0k} + \sum_{j=1}^{p} \beta_{jk} x_j} \quad \forall\, k \in \{1, \ldots, K-1\}$$

$$\mathbb{P}(Y = K|X = \mathbf{x}) = 1 - \sum_{k=1}^{K-1} \mathbb{P}(Y = k|X = \mathbf{x}) = 1 - \sum_{k=1}^{K-1} P(Y = K|X = \mathbf{x})e^{\beta_{0k} + \sum_{j=1}^{p} \beta_{jk} x_j}$$

$$\mathbb{P}(Y = K|X = \mathbf{x}) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_{0k} + \sum_{j=1}^{p} \beta_{jk} x_j}}$$

y clasificamos $\mathbf{x}$ en aquella clase k que hace máxima $\mathbb{P}(Y = k|X = \mathbf{x})$.

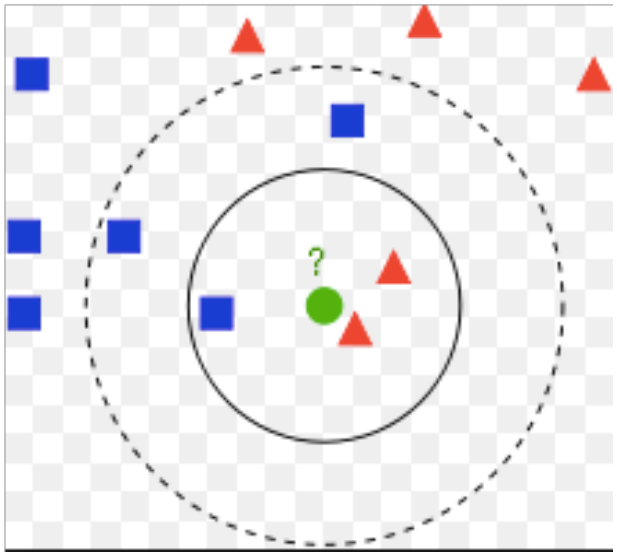
Plan

kNN. Introduction

It is a non linear classifier as SVM, CART, RF, etc. This kind of methods allow nonlinear boundary and are more flexible.

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.

We can also apply k-NN to a regression problem. In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.



- If $k = 1$ (1-nn), then the object is simply assigned to the class of its nearest neighbor. The boundary is very flexible. It is a model with low bias and high variance. and the train error equals zero but the test error rate may be quite high.
- If k grows, the model and then the boundary is less flexible, with high bias and low variance. The boundary is close to be linear.
- To avoid ties, it is better to choose an odd k.
- The method does not rely on stringent assumptions about the data
- The method works well for large n small d, but not for small n large d. For large n, the points in $N_k(\mathbf{x})$ are more likely to be close to $\mathbf{x}$. The larger d, the farther away points from each other (curse of dimensionality)
- it is often recommended to standardize the data before constructing the k-nn estimator.
- In general it uses euclidean distance, but obviously it depends of the dataset.

The k-NN classifier is defined as follow:

$$\widehat{y}(\mathbf{x}) = \frac{1}{k} \sum_{x_i \in N_k(\mathbf{x})} y_i = \text{Ave}\{y_i | \mathbf{x_i} \in N_k(\mathbf{x})\}$$

$$\widehat{f}(\mathbf{x}) = \begin{cases} 1 & \widehat{y}(\mathbf{x}) > 0{,}5 \\ 0 & \widehat{y}(\mathbf{x}) < 0{,}5 \end{cases}$$

The neighbor $N_k(\mathbf{x})$ depends obviously of a distance, and so is the prediction. Often, the classification accuracy of k-nn can be improved significantly if the distance metric (euclidean in general) is learned with specialized algorithms such as Large Margin Nearest Neighbor.

R code

```
library(class)
knn(train, test, cl, k = 1)
knn1(train, test, cl)
knn.cv(train, cl, k = 1)
```

Arguments:

- train: matrix or data frame of training set cases.
- test: matrix or data frame of test set cases.
- cl: factor of true classifications of training set
- k: number of neighbors considered.

Value (Output): Factor of classifications of the test set.

- For each row of the test set, the k nearest (in Euclidean distance) training set vectors are found, and the classification is decided by majority vote, with ties broken at random.
- If there are ties for the kth nearest vector, all candidates are included in the vote.

R code - Example

```
> head(iris3)
, , Setosa

     Sepal L. Sepal W. Petal L. Petal W.
[1,]      5.1      3.5      1.4      0.2
[2,]      4.9      3.0      1.4      0.2
[3,]      4.7      3.2      1.3      0.2
[4,]      4.6      3.1      1.5      0.2
[5,]      5.0      3.6      1.4      0.2
[6,]      5.4      3.9      1.7      0.4

, , Versicolor

     Sepal L. Sepal W. Petal L. Petal W.
[1,]      7.0      3.2      4.7      1.4
[2,]      6.4      3.2      4.5      1.5
[3,]      6.9      3.1      4.9      1.5
[4,]      5.5      2.3      4.0      1.3
[5,]      6.5      2.8      4.6      1.5
[6,]      5.7      2.8      4.5      1.3

, , Virginica

     Sepal L. Sepal W. Petal L. Petal W.
[1,]      6.3      3.3      6.0      2.5
[2,]      5.8      2.7      5.1      1.9
[3,]      7.1      3.0      5.9      2.1
[4,]      6.3      2.9      5.6      1.8
[5,]      6.5      3.0      5.8      2.2
[6,]      7.6      3.0      6.6      2.1
```

```
train = rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test = rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
cl =factor(c(rep("s",25), rep("c",25), rep("v",25)))
model=knn(train, test, cl, k = 3, prob=TRUE)
errortest=mean(model!=cl)
```

Role of k

- The smaller k, the lower bias and higher variance
- The larger k, the higher bias and lower variance (reduce the effect of noise)
- When $k = 1$, the training error is zero (overfitting)

The best choice of k depends of the data set and it is computed generally by cross-validation.

Role of k

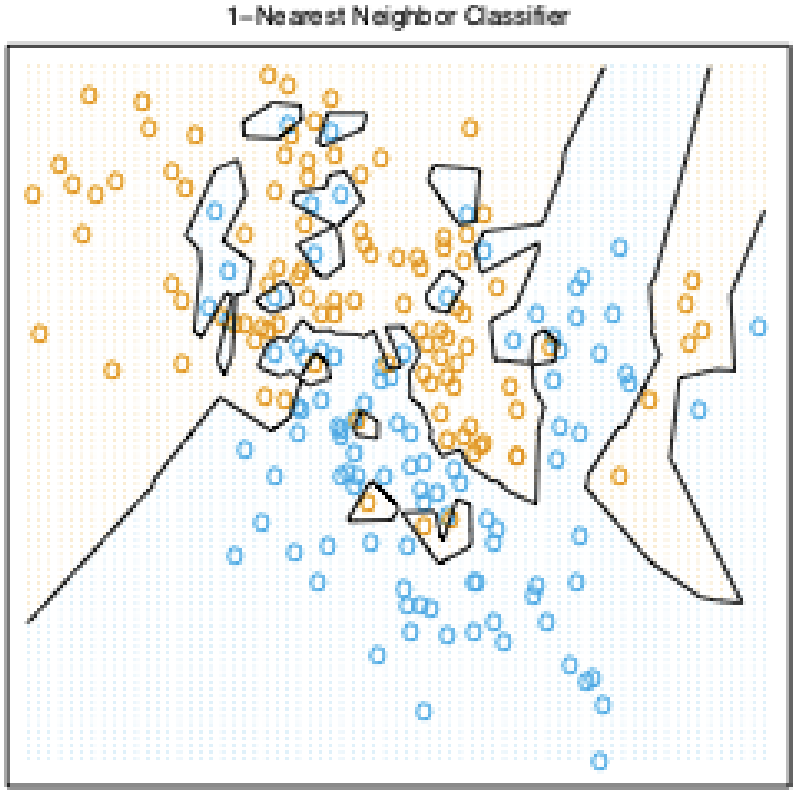


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1), and then predicted by 1-nearest-neighbor classification.*

Figura: Hastie, Tibshirani, Friedmann, *The Elements of Statistical Learning*, 2001

Role of k

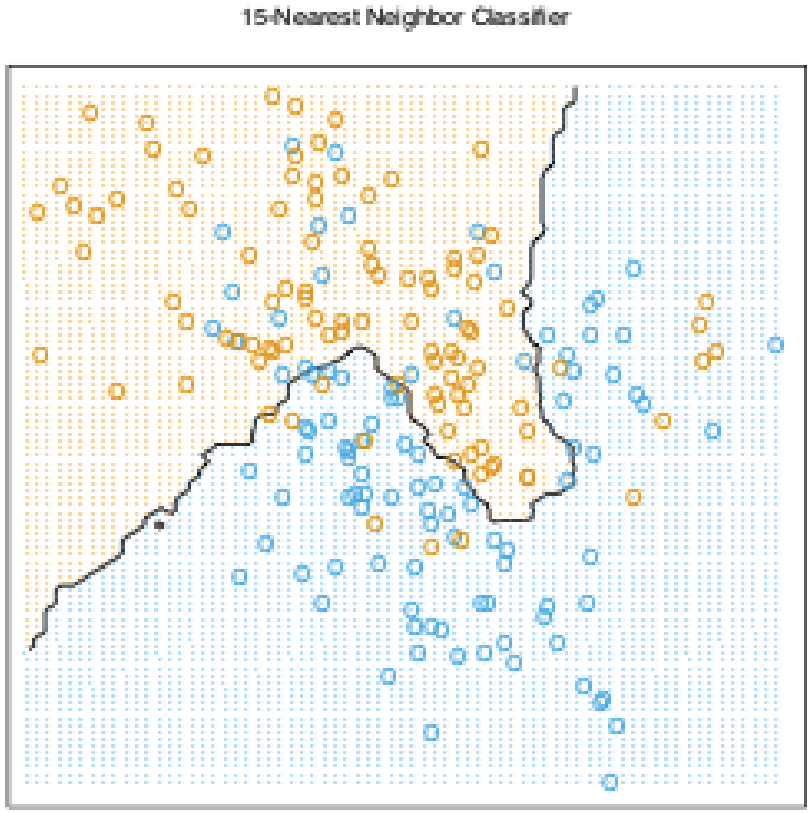


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1) and then fit by 15-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the 15-nearest neighbors.*

Figura: Hastie, Tibshirani, Friedmann, *The Elements of Statistical Learning*, 2001

Role of k

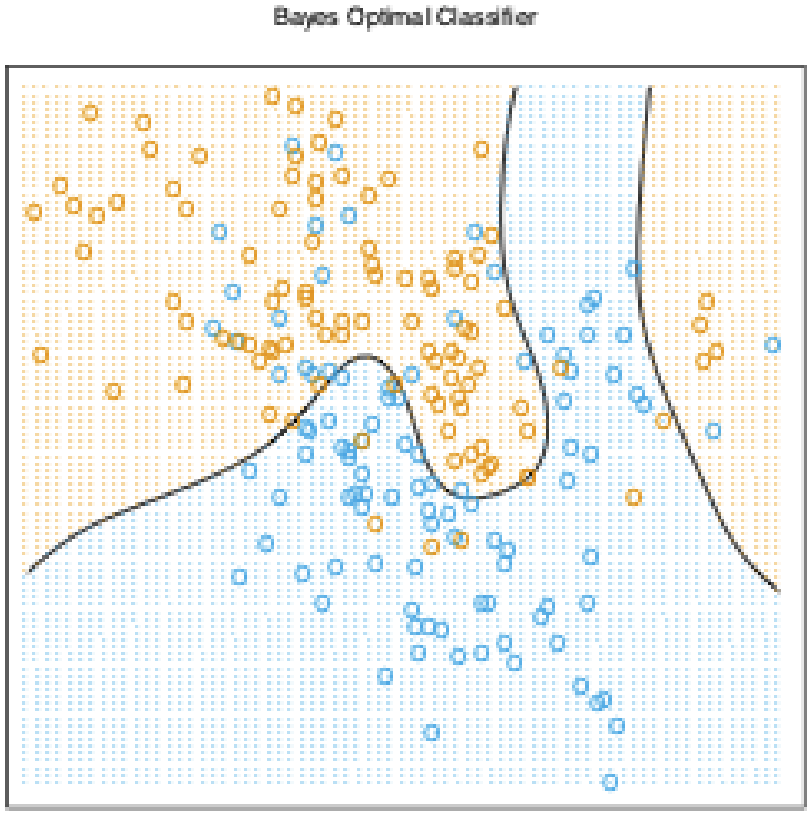


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

Figura: Hastie, Tibshirani, Friedmann, *The Elements of Statistical Learning*, 2001

Choosing k...by cross validation

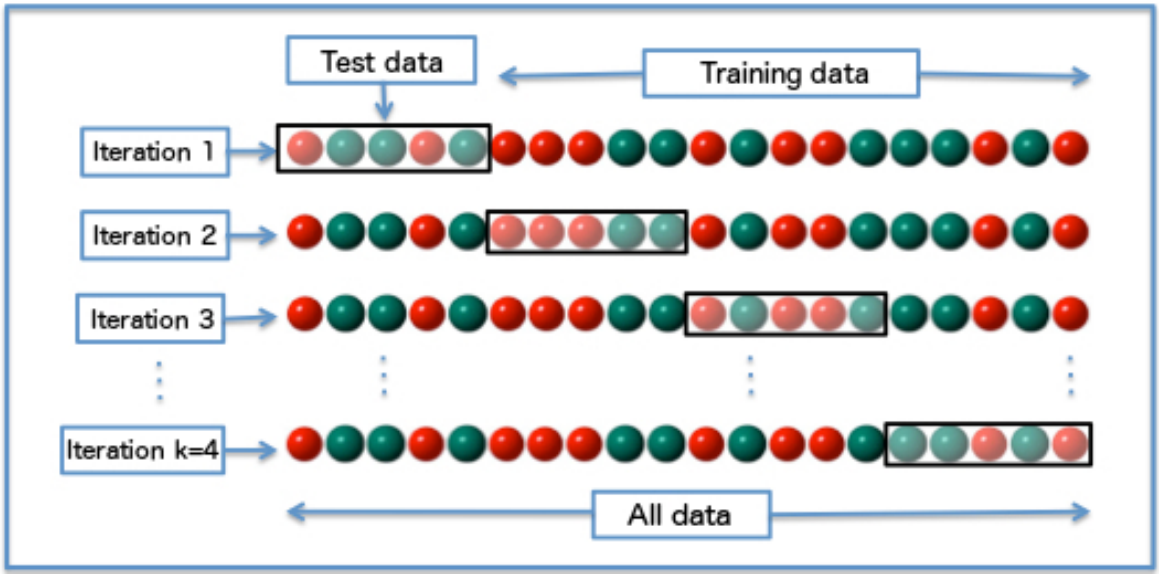


Figura: Cross Validation Scheme. Here $V = 4$.

Let $\mathcal{L} = \mathcal{L}_1 \cup \cdots \cup \mathcal{L}_V$. At each iteration we consider

$$\alpha^* = \underset{\alpha}{\text{Argmin}} \frac{1}{V} \sum_{v=1}^{V} \text{Error}\left(f_\alpha^{-v}(\mathcal{L}_v)\right)$$

where f_α^{-v} is the classifier with parameter α trained on set $\mathcal{L} \setminus \mathcal{L}_v$.
In case of k-NN, parameter α equal to the number of neighbors k.

Link with Bayes classifier

Bayes classifier assign to $\mathbf{x}$ class that maximizes posterior probability:

$$f(\mathbf{x}) = \underset{y \in \{0,1\}}{\text{Argmax}} \mathbb{P}(y|\mathbf{X} = \mathbf{x})$$

This probability can be approximated looking at the proportion of each class between the K nearest neighbor of $\mathbf{x}$, i. e

$$f(x) \approx \underset{k \in \{0,1\}}{\text{Argmax}} \frac{N_k(\mathbf{x})}{n}$$

where $N_k(\mathbf{x})$ is the number of neighbor of $\mathbf{x}$ in class k.