USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA
FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Reglas Discriminantes en el Análisis Multivariante y su adaptación a la alta dimensión

Paula Soto Rodríguez

2021/2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Reglas Discriminantes en el Análisis Multivariante y su adaptación a la alta dimensión

Paula Soto Rodríguez

Julio, 2022

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Trabajo propuesto

Área de Coñecemento: Estadística
Título: Reglas Discriminantes en el Análisis Multivariante y su adaptación a la alta dimensión
Breve descrición do contido
Se trata de estudiar algunas técnicas de clasificación o análisis discriminante, importantes y de referencia, y el papel que desempeñan dichas técnicas en el campo del "Big Data" y la alta dimensión, con sus correspondientes adaptaciones a este contexto.
Recomendacións
Guión aproximado: 1) Técnicas de clasificación lineal o cuadrática 2) Algunas técnicas de clasificación adaptadas al contexto de alta dimensión en "Big Data". 3) Ilustración en bases de datos reales
Outras observacións
Breve planificación: Revisión de las técnicas de clasificación clásica a finales de enero. Técnicas adaptadas a la alta dimensión a finales de abril. Ilustración con datos reales a lo largo del mes de mayo. El mes de junio se dedicará a ultimar la redacción final del documento.

Índice

Resumen

Este trabajo trata sobre el análisis discriminante y la introducción de algunos métodos de clasificación, tanto tradicionales como de creación más moderna. En el primer capítulo se presentan las nociones básicas de clasificación, así como dos reglas discriminantes clásicas: la lineal y la cuadrática. Además, se describen algunas generalizaciones de estas reglas, como las máquinas de vector soporte, que buscan suplir muchas de las deficiencias de los métodos más tradicionales. En el segundo capítulo se explican diversas técnicas de clasificación adaptadas al contexto del Big Data, donde la dimensión o, equivalentemente, el número de variables, es mayor que el tamaño de la muestra. Finalmente, el último capítulo ilustra el funcionamiento de estas reglas en conjuntos de datos simulados y mediante su aplicación a dos bases de datos reales.

Abstract

The aim of this project is to introduce discriminant analysis and some classification techniques, both traditional and modern. In the first chapter, basic notions for classification and two classic discrimination rules, linear and quadratic, are presented. Furthermore, this chapter also describes some generalizations of these rules, such as the support vector machines, that seek to make up for many of the deficiencies of the more traditional methods. The second chapter discusses several classifcation techniques in the Big Data context, where the dimension or, in other words, the number of variables, is bigger than the sample size. Finally, the last chapter illustrates the performance of these rules in simulated datasets and through their application on two real databases.

Introducción

El análisis discriminante es una disciplina estadística cuyo objetivo es separar los individuos de una población en distintos grupos o clases, conocidos de antemano, en base a sus características. La regla discriminante óptima será aquel criterio que mejor separe los datos.

El problema de la discriminación o clasificación se puede abordar de múltiples formas y aparece en numerosas ocasiones de la vida cotidiana, desde el diagnóstico médico hasta la detección de transacciones bancarias fraudulentas. Las técnicas presentadas en este trabajo se enmarcan dentro de la Clasificación Supervisada, donde se persigue predecir el resultado final de una cierta medida (*output*) basándose en un conjunto de valores previos (*input*). En nuestro caso, disponemos de un conjunto de variables o predictores (*inputs*) bien clasificados que sirven como modelo o referencia para la asignación de nuevas observaciones a uno u otro grupo (*output*).

El área de la estadística se enfrenta constantemente a problemas derivados de los nuevos retos que surgen con el avance de la ciencia y la tecnología. En sus comienzos, estos retos procedían, sobre todo, de la agricultura o la industria y tenían un bajo alcance. Sin embargo, con la aparición de los ordenadores y la difusión de Internet, los problemas estadísticos se han vuelto más complejos y de mayor tamaño. La enorme cantidad de datos que genera la sociedad actual ha dado paso a una nueva era de la información, conocida como Big Data, donde se necesitan herramientas computacionales no convencionales para procesar los datos adecuadamente.

Ahora bien, ¿cómo saber cuándo un conjunto de datos es o no Big Data? Los datos masivos se pueden describir por las siguientes propiedades, conocidas como las cinco V's: *volumen* (gran cantidad de datos generados y guardados), *velocidad* (rapidez a la cual se generan los datos), *variedad* (referida a los diversidad de los datos: estructurados, semi-estructurados y no estructurados), *veracidad* (el grado de fiabilidad en los datos debe ser alto, con resultados de calidad y verificables) y *valor* (los datos generados deben ser útiles y accionables, es decir, deben ayudar a tomar una decisión en base a ellos).

La evolución de la tecnología ha originado, a su vez, el florecimiento de numerosas disciplinas estadísticas. De todas ellas cabe destacar el análisis de datos, cuya finalidad es reconocer patrones e inferir conclusiones en los datos no procesados (*raw data*) para extraer información útil que

sirva de apoyo en la toma de decisiones.

Un aspecto importante en cualquier análisis de datos es la interpretación, y uno puede preguntarse: ¿Qué información proporciona el análisis acerca de los datos? ¿Qué nuevos conocimientos hemos adquirido tras el análisis? ¿Cuán adecuado es el método escogido para esos datos? ¿Cuáles son las limitaciones de un determinado método? y ¿Qué otros métodos producirían mejores resultados? La necesidad de responder a todas estas preguntas, así como de afrontar los problemas que aparecen en el contexto de la alta dimensión, ha traído consigo nuevas formas de clasificación para separar de forma óptima los datos.

En este trabajo se presentan algunas de tales reglas discriminantes que intentan lidiar con los dificultades surgidas cuando el número de variables es mucho mayor que el tamaño de la muestra, aplicando métodos dispersos con pocos parámetros que reduzcan la dimensión y, al mismo tiempo, cuenten con un elevado poder de predicción. Además, se estudia su comportamiento sobre bases de datos simuladas y reales, ilustrando las ventajas y carencias de los métodos y comparando su eficiencia con la de los clasificadores tradicionales.

Capítulo 1

Técnicas de clasificación lineal y cuadrática

Según la RAE, discriminar significa "seleccionar excluyendo". Dada una muestra dividida en k grupos, la definición anterior se traduce matemáticamente por construir una regla (discriminante) que mejor separe los datos de los grupos. La aplicación de dicha regla se denomina clasificación (creación de clases). En este capítulo estudiaremos en profundidad dos de ellas: la lineal y la cuadrática. Además, introduciremos otros métodos de clasificación más actuales que solventarán muchas de sus deficiencias, consiguiendo, con ello, una mejor separación de los datos.

1.1. Definiciones básicas

Dada una muestra dividida en k grupos, los objetivos del Análisis Discriminante son:

- Obtener un criterio para asignar un nuevo individuo a uno u otro grupo.
- Determinar la estructura diferenciadora (discriminante) de los grupos o, equivalentemente, qué combinación lineal de las variables originales separa mejor los datos.

Es importante notar que no pretendemos clasificar a los individuos en grupos procediendo nosotros mismos a su agrupación, sino que aquí los grupos son conocidos de antemano y lo que buscamos es descubrir qué tiene de específico cada grupo para ser capaces de asignar de manera correcta un nuevo individuo a uno de ellos.

A continuación, establecemos la notación que se seguirá a lo largo de todo el trabajo, así como algunos conceptos fundamentales. Tomaremos como referencia [8].

Definición 1.1. Sea $k \geq 2$ (fijado y conocido) el número de clases y sean $\mathcal{C}_1, ..., \mathcal{C}_k$ las clases. Un vector d-dimensional $\mathbf{X}$ pertenece a la clase $\mathcal{C}_\nu$ o $\mathbf{X}$ es un miembro de la clase $\mathcal{C}_\nu$ para algún $\nu \leq k$, si $\mathbf{X}$ satisface las propiedades que caracterizan $\mathcal{C}_\nu$. Lo denotamos como: $\mathbf{X} \in \mathcal{C}_\nu$ o $\mathbf{X}^{[\nu]}$.

Para vectores aleatorios $\mathbf{X}$ de k clases, la etiqueta Y de $\mathbf{X}$ es una variable aleatoria que toma un conjunto discreto de valores 1 ,..., k tal que: $Y = \nu$ si $\mathbf{X} \in \mathcal{C}_\nu$.

El vector aleatorio d+1 dimensional etiquetado se denota por:

$$\begin{bmatrix} \mathbf{X} \\ \mathrm{Y} \end{bmatrix}$$

Definición 1.2. Sean $\mathcal{C}_1, ..., \mathcal{C}_k$ k clases. Sea $\mathbf{X}$ un vector aleatorio que pertenece a una de estas clases. Una regla (discriminante) o clasificador $\mathfrak{r}$ para $\mathbf{X}$ es una aplicación que asigna $\mathbf{X}$ un número $l \leq k$. Escribimos: $\mathfrak{r}(\mathbf{X}) = l$ para $1 \leq l \leq k$.

La regla $\mathfrak{r}$ asigna $\mathbf{X}$ a la clase correcta o clasifica $\mathbf{X}$ correctamente si: $\mathfrak{r}(\mathbf{X}) = \nu$ cuando $\mathbf{X}$ pertenece a la clase $\mathcal{C}_\nu$ y clasifica $\mathbf{X}$ incorrectamente en otro caso.

Definición 1.3. Sea $\mathbf{X}$ un vector aleatorio que pertenece a una de las dos clases $\mathcal{C}_1$ o $\mathcal{C}_2$. Sea $\mathfrak{r}$ la regla discriminante para $\mathbf{X}$. Una función de decisión para $\mathbf{X}$, asociada con $\mathfrak{r}$, es una función escalar h tal que:

$$h(\mathbf{X}) = \begin{cases} > 0 & \text{si } \mathfrak{r}(\mathbf{X}) = 1 \\ < 0 & \text{si } \mathfrak{r}(\mathbf{X}) = 2 \end{cases}$$

La función de decisión correspondiente a una regla no es única. Por ejemplo, cualquier múltiplo de una función de decisión por un escalar positivo es también una función de decisión para la misma regla.

Idealmente, el número asignado a $\mathbf{X}$ por la regla $\mathfrak{r}$ sería el mismo que el valor de su etiqueta. Sin embargo, en la práctica esto no siempre ocurre, lo que conduce a comparaciones en la efectividad de las distintas reglas.

1.2. Análisis Discriminante Lineal

El análisis discriminante lineal (LDA) es un método de clasificación supervisado de variables cualitativas (i.e, que toman valores en un conjunto discreto) en el que dos o más grupos son conocidos *a priori* y las nuevas observaciones se clasifican en uno de ellos en función de sus características.

Existen dos formas de ver el LDA:

- Geométrica, haciendo uso de una reducción de dimensiones. Este fue el primer enfoque del análisis discriminante lineal.
- Analítica, empleando probabilidades y el Teorema de Bayes.

1.2.1. LDA y la reducción de dimensiones

El origen del análisis discriminante lineal se remonta al matemático R. Fisher [5]. Su propuesta consistía en hallar la dirección $\mathbf{e}$ que maximizara la diferencia entre grupos respecto a la variabilidad dentro de los grupos y trabajar con estas cantidades unidimensionales.

Es decir, Fisher planteó una aproximación en la que el espacio p-dimensional (donde p es el número de predictores originales) se reduce a un subespacio de menos dimensiones formado por las combinaciones lineales de las variables originales que mejor explican la separación de las clases. Una vez encontradas dichas combinaciones, se realiza la clasificación en este subespacio. El subespacio óptimo, según Fisher, es aquel que maximiza la distancia entre grupos en términos de varianza.

La aproximación de Fisher se puede ver, por tanto, como un proceso con dos partes:

- Reducción de dimensionalidad: Se pasa de p variables predictoras originales a l combinaciones lineales de dichos predictores (variables discriminantes) que permiten explicar la separación de los grupos pero con menos dimensiones ($l<p$). Por ello se dice que el LDA funciona como un reductor de dimensiones.
- Clasificación de las observaciones empleando las variables discriminantes.

Matemáticamente, el problema de discriminación de Fisher es equivalente al siguiente problema de optimización:

$$\begin{aligned} \text{máx} \quad & \frac{\mathbf{e}^T B \mathbf{e}}{\mathbf{e}^T W \mathbf{e}} \\ \text{sujeto a:} \quad & \mathbf{e}^T \mathbf{e} = 1 \end{aligned} \tag{1.1}$$

donde $\mathbf{e}$ es un vector d-dimensional, $\mathrm{B} = \sum_{\nu=1}^{k} (\boldsymbol{\mu_\nu} - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu_\nu} - \bar{\boldsymbol{\mu}})^T$ es la variabilidad entre grupos y $\mathrm{W} = \sum_{\nu=1}^{k} \Sigma_\nu$ es la variabilidad intra-grupos. Resolver el problema 1.1 se corresponde con encontrar la descomposición en autovalores y autovectores de la matriz: $\mathrm{W}^{-1}\mathrm{B}$. La dirección $\boldsymbol{\eta}$ que produce la máxima separación entre los grupos es el mayor autovalor de $\mathrm{W}^{-1}\mathrm{B}$ y se denomina dirección discriminante.

Además, se tiene:

- Los autovectores proporcionan las combinaciones lineales discriminantes (funciones discriminantes), que son incorreladas entre ellas.
- Los autovalores nos dan la medida de la capacidad discriminante.

Definición 1.4. Para $\nu \leq k$, sea $\mathcal{C}_\nu$ clases caracterizadas por (μ_ν, Σ_ν). Sea $\mathbf{X}$ un vector aleatorio de una de las clases $\mathcal{C}_\nu$. Consideremos las matrices B y W definidas anteriormente y asumamos que W es invertible. Sea $\boldsymbol{\eta}$ la dirección discriminante.
La regla (lineal) discriminante de Fisher $\mathfrak{x}_F$ se define como:

$$\mathfrak{x}_F(\mathbf{X}) = l \quad si \ |\boldsymbol{\eta}^T\boldsymbol{X} - \boldsymbol{\eta}^T\boldsymbol{\mu_l}| < |\boldsymbol{\eta}^T\boldsymbol{X} - \boldsymbol{\eta}^T\boldsymbol{\mu_\nu}| \qquad \text{para todo } \nu \neq l \tag{1.2}$$

Es decir, la regla de Fisher asigna $\mathbf{X}$ el número l si el escalar $\boldsymbol{\eta}^T\mathbf{X}$ está más cerca de la media escalar $\boldsymbol{\eta}^T\boldsymbol{\mu_l}$. Por tanto, en vez de fijarse en la verdadera media $\boldsymbol{\mu_l}$ que está mas cerca de $\mathbf{X}$, escogemos la cantidad más sencilla: $\boldsymbol{\eta}^T\boldsymbol{\mu_l}$, la cual está más próxima a media ponderada $\boldsymbol{\eta}^T\mathbf{X}$. Usar el escalar $\boldsymbol{\eta}^T\mathbf{X}$ presenta varias ventajas frente al vector $\mathbf{X}$ d-dimensional:

- Reduce y simplifica las comparaciones multivariantes, pasándolas a una dimensión.
- Disminuye el efecto de variables explicativas irrelevantes.

Para dos clases con medias $\boldsymbol{\mu}_1$ y $\boldsymbol{\mu}_2$, una función de decisión h adoptaría la forma:

$$h(\mathbf{X}) = \left[\mathbf{X} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)\right]^T \boldsymbol{\eta} \tag{1.3}$$

Notar que, en efecto, h es una función de decisión para $\mathfrak{x}_F$, pues $h(\mathbf{X}) > 0 \Longleftrightarrow \mathfrak{x}_F(\mathbf{X}) = 1$ y $h(\mathbf{X}) < 0 \Longleftrightarrow \mathfrak{x}_F(\mathbf{X}) = 2$

1.2.2. LDA y la regla de Bayes

Haciendo uso del teorema de Bayes, LDA estima la probabilidad de que una observación, dado un determinado valor de los predictores, pertenezca a cada una de las clases de la variable cualitativa, i.e, estima: $\mathbb{P}(Y = \nu | X = x)$. Finalmente, asigna la observación a la clase ν para la cual la probabilidad predicha es mayor. Veamos esto con detalle.

Teorema 1.5 (Teorema de Bayes). *Sea $\{\mathcal{A}_1, ..., \mathcal{A}_i, ..., \mathcal{A}_n\}$ un conjunto de sucesos mutuamente excluyentes y exhaustivos tales que la probabilidad de cada uno de ellos es distinta de cero ($\mathbb{P}[\mathcal{A}_i] \neq 0$, para $i = 1, 2, ...n$). Si $\mathcal{B}$ es un suceso cualquiera del que se conocen las probabilidades condicionales $\mathbb{P}(\mathcal{B}|\mathcal{A}_i)$, entonces la probabilidad $\mathbb{P}(\mathcal{A}_i|\mathcal{B})$ viene dada por la expresión:*

$$\mathbb{P}(\mathcal{A}_i|\mathcal{B}) = \frac{\mathbb{P}(\mathcal{B}|\mathcal{A}_i)\mathbb{P}(\mathcal{A}_i)}{\mathbb{P}(\mathcal{B})}$$

donde:

- $\mathbb{P}(\mathcal{A}_i)$ *son las probabilidades a priori.*
- $\mathbb{P}(\mathcal{B}|\mathcal{A}_i)$ *es la probabilidad de* $\mathcal{B}$ *en la hipótesis* $\mathcal{A}_i$.
- $\mathbb{P}(\mathcal{A}_i|\mathcal{B})$ *son las probabilidades a posteriori.*

Teniendo en cuenta el anterior teorema y la Ley de probabilidad total se obtiene la Fórmula de Bayes, también conocida como Regla de Bayes (o regla de clasificación universal):

$$\mathbb{P}(\mathcal{A}_i|\mathcal{B}) = \frac{\mathbb{P}(\mathcal{B}|\mathcal{A}_i)\mathbb{P}(\mathcal{A}_i)}{\sum_{k=1}^{n} \mathbb{P}(\mathcal{B}|\mathcal{A}_k)\mathbb{P}(\mathcal{A}_k)}$$

Supongamos ahora que deseamos clasificar una nueva observación en una de las ν clases de una variable cualitativa Y, siendo el número de clases $k \geq 2$. Consideremos un único predictor X (i.e, p=1) En otras palabras, Y puede tomar ν posibles valores. Sea π_ν la probablidad *a priori* de que una observación, aleatoriamente escogida, pertenezca a la ν-ésima clase. Denotemos por $f_\nu(X) = \mathbb{P}(X|Y = \nu)$ la función de densidad de X para una observación que pertenezca a la ν-ésima clase. Equivalentemente, $f_\nu(X)$ es relativamente grande si hay una alta probabilidad de que una observación de la ν-ésima clase verifique: X $\approx x$.

Entonces, el Teorema de Bayes establece que:

$$\mathbb{P}(Y = \nu|X = x) = \frac{\pi_\nu f_\nu(X)}{\sum_{l=1}^{j} \pi_l f_l(X)} \tag{1.4}$$

En consonancia con nuestra notación anterior, usaremos $p_\nu(x) = \mathbb{P}(Y = \nu|X = x)$ para designar la probabilidad *a posteriori* de que una observación X $= x$ pertenezca a la ν-ésima clase. Esto es, la probabilidad de que una observación pertenezca a la ν-ésima clase, *dado* el valor del predictor para esa observación.

Teniendo todo esto en cuenta, la regla discriminante óptima (i.e, con menor error de clasificación) consistirá en asignar la observación a aquel grupo que maximice la probabilidad a

posteriori. Dado que el denominador $\sum_{l=1}^{j} \pi_l f_l(X)$ es igual para todas las clases (puesto que no depende de ν), la norma de clasificación es equivalente a decir que se asignará cada observación a aquel grupo para el que $\pi_i f_i(X), i = 1, ...k$ sea mayor.

Para que la clasificación basada en Bayes sea posible, se necesitan conocer π_ν y $f_\nu(X) = \mathbb{P}(X|Y = k)$. En la práctica, rara vez se dispone de esta información, por lo que los parámetros tienen que ser estimados a partir de la muestra. El clasificador LDA es el resultado de sustituir los parámetros π_ν y $f_\nu(X)$ por sus correspondientes análogos muestrales en el clasificador de Bayes.

En general, estimar π_ν es sencillo si tenemos una muestra aleatoria de la población: basta con calcular la fracción de las observaciones que pertenecen a la ν-ésima clase. Sin embargo, estimar la función de densidad $f_\nu(X)$ es más complejo, por lo que es necesario asumir ciertas hipótesis adicionales sobre $f_\nu(X)$, como, por ejemplo, que los datos sigan una distribución normal o gaussiana. Esto nos conduce a la siguiente sección.

1.2.3. LDA para poblaciones normales

LDA es óptimo (según la regla de Bayes) bajo la suposición de que los datos son normales y homocedásticos, esto es, los grupos son normales multivariantes que solo se diferencian por sus medias.

Obtendremos la regla (lineal) normal discriminante (designada por algunos autores como regla de Fisher) considerando un único predictor y dos clases, i.e, $p = 1$ y $k = 2$.

Asumamos que $f_\nu(X)$ es normal. Si X es una variable aleatoria unidimensional, su función de densidad es de la forma:

$$f_\nu(X) = \frac{1}{\sigma_\nu \sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma_\nu^2}(x - \mu_\nu)^2\right)$$

donde μ_ν y σ_ν^2 son la media y la varianza de la ν-ésima clase, respectivamente. Si asumimos también que los datos son homomcedásticos, i.e, que las ν clases tiene una varianza común: $\sigma_1^2 = ... = \sigma_\nu^2 = \sigma^2$, se tiene, sustituyendo $f_\nu(X)$ en 1.4:

$$p_\nu(x) = \frac{\pi_\nu \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_\nu)^2\right)}{\sum_{l=1}^{j} \pi_l \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_l)^2\right)}$$

Tomando logaritmos en la expresión anterior y reordenando términos, obtenemos:

$$\delta_\nu(x) = \log(\mathbb{P}(Y = \nu|X = x)) = x \cdot \frac{\mu_\nu}{\sigma^2} - \frac{\mu_\nu^2}{2\sigma^2} + \log(\pi_\nu), \quad \text{para } \nu = 1, 2 \tag{1.5}$$

Si asumimos que $\pi_1 = \pi_2$ y que $\mu_1 > \mu_2$, la regla discriminante lineal óptima, también conocida como clasificador de Bayes, se define entonces como:

$$\mathfrak{r}_{Bayes}(\mathrm{X}) = 1 \quad \text{si } \delta_1(x) > \delta_2(x) \Longleftrightarrow \mathfrak{r}_{Bayes} = \arg\max_\nu \left\{\log \pi_\nu + \mu_\nu^T \Sigma^{-1}(\mathbf{X} - \mu_\nu)\right\} \tag{1.6}$$

Equivalentemente, la regla de Bayes asigna una nueva observación a la clase 1 si:

$$\mathfrak{r}_{Bayes}(\mathbf{X}) = \mathbb{I}\left(\mathbf{X} - \frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) + \log\frac{\pi_1}{\pi_2} > 0\right) \tag{1.7}$$

donde $\mathbb{I}(z > 0)$ es la función indicadora o característica, que toma el valor 1 si $z > 0$ y 0 en otro caso (vemos por tanto que 1.7 es lo mismo que: $\mathfrak{r}_{Bayes}(\mathrm{X}) = 1$ si $\delta_1(x) > \delta_2(x) \Longleftrightarrow x \in \mathcal{C}_1$).

El término lineal en el nombre del clasificador deriva del hecho de que las funciones discriminantes $\delta_\nu(x)$ son funciones lineales de x.

El resultado obtenido para una sola variable se generaliza fácilmente para el caso multidimensional.

Teorema 1.6. *Sea $\boldsymbol{X}$ un vector aleatorio con distribución normal que pertenece a una de las clases $\mathcal{C}_\nu = \mathcal{N}(\mu_\nu, \Sigma)$, para $\nu = 1, 2$ y asumamos que $\mu_1 \neq \mu_2$. Sea δ_ν el logaritmo de la probabilidad a posteriori de que que una observación $X = x$ pertenezca a la ν-ésima clase. Sea $\mathfrak{r}_{Bayes}$ la regla que asigna $\boldsymbol{X}$ a la clase 1 si $\delta_1 > \delta_2$. Entonces:*

$$h(\boldsymbol{X}) = \left[\boldsymbol{X} - \frac{1}{2}(\boldsymbol{\mu_1} + \boldsymbol{\mu_2})^T \Sigma^{-1}(\boldsymbol{\mu_1} - \boldsymbol{\mu_2})\right] \tag{1.8}$$

es una función de decisión para $\mathfrak{r}_{Bayes}$, y $h(\boldsymbol{X}) > 0$ si y solo si $\delta_1(\boldsymbol{X}) > \delta_2(\boldsymbol{X})$.

El clasificador LDA se obtiene sustituyendo $\Sigma, \mu_1, \mu_2, \pi_1$ y π_2 por sus respectivos estimadores muestrales ($\nu = 1, 2$):

$$\hat{\pi}_\nu = n_\nu/n, \quad \hat{\boldsymbol{\mu}}_\nu = \sum_{Y_i = \nu} \mathbf{X}_i/n_\nu, \quad \hat{\Sigma} = \sum_\nu \sum_{Y_i = \nu} (\mathbf{X}_i - \hat{\boldsymbol{\mu}}_\nu)(\mathbf{X}_i - \hat{\boldsymbol{\mu}}_\nu)^T/(n - 2)$$

donde n es el tamaño muestral y n_ν es el número de observaciones en la ν-ésima clase.

Entonces, la regla LDA se define como:

$$\mathfrak{x}_{LDA}(\mathbf{X}) = \mathbb{I}\left(\mathbf{X} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2)^T\hat{\Sigma}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2) + \log\frac{\hat{\pi}_1}{\hat{\pi}_2} > 0\right) \tag{1.9}$$

1.3. Análisis Discriminante Cuadrático

Las reglas discriminantes discutidas hasta el momento eran lineales en el vector aleatorio $\mathbf{X}$. En otras palabras, sus funciones de decisión adoptaban la forma:

$$h(\mathbf{X}) = \mathbf{a}^T\mathbf{X} + c$$

para algún vector $\mathbf{a}$ y escalar c que no dependen de $\mathbf{X}$.

En la regla normal discriminante, la linealidad era consecuencia de que todas las clases tenían la misma matriz de varianzas-covarianzas. Si se relaja la hipótesis de homocedasticidad, la regla de Bayes nos lleva a una función de decisión no lineal y, en consecuencia, a una regla no lineal. En concreto, nos conduce al Análisis Discriminante Cuadrático (QDA).

Al igual que LDA, el clasificador QDA resulta de asumir que las observaciones de cada clase siguen una distribución gaussiana, pero a diferencia de éste, QDA supone que cada clase tiene su propia matriz de varianzas-covarianzas. Esto es, tiene como hipótesis que una observación de la ν-ésima clase es de la forma $\mathrm{X} \sim \mathcal{N}(\mu_\nu, \Sigma_\nu)$. Bajo todas estas asunciones, el clasificador de Bayes asigna una observación $X = x$ al grupo para el cual:

$$\begin{aligned}\delta_\nu(x) &= -\frac{1}{2}(x - \boldsymbol{\mu}_\nu)^T(\Sigma_\nu)^{-1}(x - \boldsymbol{\mu}_\nu) - \frac{1}{2}\log|\Sigma_\nu| + \log(\pi_\nu) \\ &= -\frac{1}{2}x^T(\Sigma_\nu)^{-1}x + x^T(\Sigma_\nu)^{-1}\boldsymbol{\mu}_\nu - \frac{1}{2}\boldsymbol{\mu}_\nu^T(\Sigma_\nu)^{-1}\boldsymbol{\mu}_\nu - \frac{1}{2}\log|\Sigma_\nu| + \log(\pi_\nu)\end{aligned} \tag{1.10}$$

es mayor. Al contrario que en el LDA, en este caso x aparece como una función cuadrática. De ahí viene el nombre de Análisis Discriminante Cuadrático.

Por tanto, en el contexto binario, la regla QDA asigna una nueva observación a la clase 1 si:

$$\mathfrak{x}_{QDA}(\mathbf{X}) = \mathbb{I}\left((\mathbf{X} - \mu_1)^T\Sigma_1^{-1}(\mathbf{X} - \mu_1) - (\mathbf{X} - \mu_2)^T\Sigma_2^{-1}(\mathbf{X} - \mu_2) + \log\frac{|\Sigma_1|}{|\Sigma_2|} > 0\right) \tag{1.11}$$

1.4. LDA vs. QDA: ¿cuál es mejor?

Llegado este momento cabe preguntarse cuándo se debería utilizar uno u otro método, y si alguno de los dos presenta más ventajas frente al otro. La respuesta a estas cuestiones viene de considerar el balance sesgo-varianza.

En la práctica, los parámetros poblacionales se desconocen y tienen que ser estimados por una muestra de entrenamiento, así como la media y la matriz de varianzas-covarianzas. Cuando tenemos p variables, estimar una matriz de covarianzas requiere estimar $\frac{p(p+1)}{2}$ parámetros. Por tanto, QDA necesita $k \cdot \frac{p(p+1)}{2}$, pues estima una matriz de varianzas-covarianzas separada para cada clase. Si asumimos que los grupos poseen una varianza común, como en el caso del LDA, solo sería necesario aproximar $k \cdot p$ coeficientes. En consecuencia:

- El número de parámetros estimados en LDA crece linealmente con p, mientras que en QDA se incrementan de forma cuadrática. Por tanto, esperaríamos que QDA se comporte peor que LDA en problemas de altas dimensiones.
- LDA es un clasificador mucho menos flexible que QDA, lo que implica que tiene una varianza substancialmente inferior. Esto conduce a un mayor poder de predicción.
- Si los grupos no comparten la misma matriz de varianzas-covarianzas, el clasificador LDA tendría un elevado sesgo, acarreando un deficiente rendimiento. En esencia, LDA es mejor que QDA si la muestra es pequeña, ya que en este caso reducir la varianza es decisivo.

En conclusión, ninguno de los dos clasificadores domina completamente al otro. En cualquier situación, la elección de uno u otro método dependerá de la distribución de las variables en cada una de las k clases, así como del número total de observaciones (n) y de predictores, i.e, de la dimensión (p).

1.5. Generalización del LDA y el QDA: RDA

El RDA, o Análisis Discriminante Regularizado, constituye un nexo de unión entre el LDA y el QDA y se fundamenta en regularizar, de forma individual, las matrices de varianzas-covarianzas. Regularizar significa imponer una cierta restricción en los parámetros estimados y es útil cuando tenemos pocos datos, pues las matrices de varianzas- covarianzas presentan mucho sesgo. En esta situación, los autovalores son mal aproximados, lo cual ocasiona problemas de inestabilidad relacionados con la inversión de las matrices.

Consideremos las clases $\mathcal{C}_1, ..., \mathcal{C}_k$, y sea $\hat{\Sigma}_\nu$ la matriz de covarianzas muestral correspondiente a la clase $\mathcal{C}_\nu$. Para $\alpha \in [0,1]$, la matriz de covarianzas regularizada $\hat{\Sigma}_\nu(\alpha)$ es:

$$\hat{\Sigma}_\nu(\alpha) = \alpha\hat{\Sigma}_\nu + (1-\alpha)\hat{\Sigma}_{pool}$$

donde $\hat{\Sigma}_{pool}$ es la matriz de varianza agrupada, obtenida mediante *pooling*, un método para estimar la varianza de un conjunto de poblaciones cuando la media de cada grupo puede ser diferente, y las verdaderas matrices de varianzas-covarianzas son desconocidas, pero se pueden asumir iguales. La matriz de varianza agrupada es una media de las matrices de covarianzas muestrales, cuya expresión viene dada por:

$$\hat{\Sigma}_{pool} = \sum_{\nu=1}^{k} \frac{n_\nu - 1}{n-2}\hat{\Sigma}_\nu$$

donde n_ν es el tamaño de la clase ν-ésima y n $= \sum_{\nu=1}^{k} n_\nu$. Bajo el supuesto de varianzas poblacionales iguales, la varianza muestral agrupada proporciona una estimación de la varianza con precisión más alta que las varianzas muestrales individuales.

Los dos casos especiales ($\alpha = 0$ y $\alpha = 1$) corresponden a la discriminación lineal y cuadrática, respectivamente. Las matrices de covarianzas distintas en la regla cuadrática se contraen a una matriz de covarianzas común, como en LDA. EL parámetro α proporciona una continuidad entre los modelos LDA y QDA y necesita ser especificado. En la práctica, α puede ser escogido basado en el comportamiento de la regla en la muestra de validación, o mediante validación cruzada.

Un segundo parámetro de penalización $\gamma \in [0,1]$, regula la contracción, a su vez, de la matriz $\hat{\Sigma}_{pool}$ hacia la matriz identidad, I, mediante:

$$\hat{\Sigma}_{pool}(\gamma) = (\gamma)\hat{\Sigma}_{pool} + (1-\gamma)s^2\,\mathrm{I}$$

donde s^2 es un escalar positivo adecuadamente elegido. La substitución de $\hat{\Sigma}_{pool}$ por $\hat{\Sigma}_{pool}(\gamma)$ conduce a una familia más genérica de matrices de varianzas covarianzas $\hat{\Sigma}_\nu(\alpha,\gamma)$, indexadas por un par de parámetros.

1.6. Evaluación de las reglas. Probabilidad de clasificación incorrecta

1.6.1. Fronteras y regiones discriminantes

Definición 1.7. Sea $\mathbf{X}$ un vector aleatorio que pertenece a una de dos clases $\mathcal{C}_1$ y $\mathcal{C}_2$. Sea $\mathfrak{x}$ una regla discriminante para $\mathbf{X}$, y sea h la correspondiente función de decisión.

1. La frontera de decisión B de la regla $\mathfrak{x}$ consiste en todos los vectores d-dimensionales $\mathbf{X}$ tal que $h(\mathbf{X})= 0$.

2. Para $\nu \leq k$, la región discriminante G_ν de la regla $\mathfrak{x}$ está definida por:

$$G_\nu = \{\boldsymbol{X} : \mathfrak{x}(\boldsymbol{X}) = \nu\}$$

Cuando la regla es lineal en $\mathbf{X}$, la función de decisión es una línea recta en 2 dimensiones o un hiperplano en d dimensiones. Para reglas no lineales surgen fronteras más complejas. Las fronteras y funciones de decisión existen para más de dos clases, pero son menos útiles debido a su complicada interpretación.

Las regiones discriminantes G_ν son disjuntas, ya que cada $\mathbf{X}$ es asignado un número $\nu \leq k$ por la regla discriminante. Podemos interpretar las regiones discriminantes como "clases asignadas por la regla". Dadas regiones disjuntas, definimos una regla discriminante como:

$$\mathfrak{x}(\boldsymbol{X}) = \nu, \qquad \text{si} \quad \boldsymbol{X} \in G_\nu$$

por lo que los dos conceptos, regiones y regla, quedan completamente determinados conocido uno de ellos. Si la regla fuese *perfecta*, entonces para cada ν, la clase $\mathcal{C}_\nu$ y la región G_ν coincidirían. Las fronteras de decisión separan las regiones discriminantes.

La Figura 1.1 ilustra el concepto de fronteras y regiones discriminantes para las reglas de Bayes, LDA y QDA, así como el comportamiento de estos métodos cuando las clases tienen una varianza común y en el caso en que las matrices de covarianzas difieren. En la primera situación, LDA se aproxima mejor a la regla de Bayes (puesto que la frontera de ésta última es lineal), resultando en un mejor clasificador. En el segundo caso, en cambio, la frontera de Bayes es cuadrática, por lo que QDA es el que la aproxima mejor.

1.6.2. Evaluación de las reglas discriminantes

Existen numerosas razones por las que uno está interesado en evaluar cuán bien funciona una regla discriminante. Si disponemos de dos o más reglas, es importante entender qué técnica es mejor y bajo qué condiciones una regla proporciona buenos resultados. En la práctica, no hay una única medida o mejor técnica de evaluación.

Sea $\mathbb{X} = [\mathbf{X}_1\mathbf{X}_2...\mathbf{X}_n]$ los datos pertenecientes a las clases $\mathcal{C}_\nu$, con $\nu \leq k$. Sea $\mathfrak{x}$ una regla discriminante para este conjunto de datos. Decimos que la regla $\mathfrak{x}$ ha clasificado correctamente una observación si la asignación de la regla coincide con el valor de la etiqueta. Cuántas más clases haya, más errores puede cometer dicha regla.

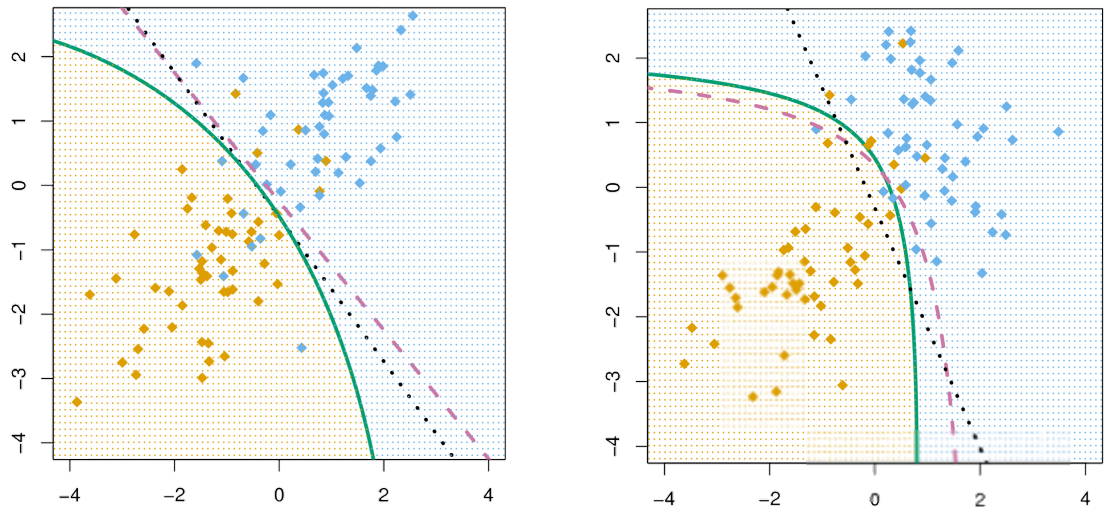

Figura 1.1: Fronteras de decisión para la regla de Bayes (línea violeta discontinua), LDA (línea negra punteada) y QDA (línea verde sólida). En sombreado se representa la región discriminante de QDA. La gráfica de la izquierda muestra un problema de clasificación binario donde $\Sigma_1 = \Sigma_2$. Como la frontera de Bayes es lineal, es mejor aproximada por LDA que QDA, por lo que en este caso LDA es un mejor clasificador que QDA. En la figura de la derecha, $\Sigma_1 \neq \Sigma_2$. Como ahora la frontera de Bayes no es lineal, QDA la aproxima mejor que LDA. Imágenes extraídas de [7]

En esta sección, consideraremos dos reglas de evaluación para medir el rendimiento de las reglas discriminantes: el error simple de clasificación y el error *dejando uno fuera*.

Definición 1.8. Sea $\mathbb{X} = [\mathbf{X}_1\mathbf{X}_2...\mathbf{X}_n]$ datos etiquetados, y sea $\mathfrak{x}$ la regla. Un factor de coste $\boldsymbol{c} = [c_1, ..., c_n]$ asociado a la regla $\mathfrak{x}$ está definido por:

$$c_i = \begin{cases} = 0 & \text{si } \mathbf{X}_i \text{ ha sido correctamente clasificado por } \mathfrak{x} \\ > 0 & \text{si } \mathbf{X}_i \text{ ha sido incorrectamente clasificado por } \mathfrak{x} \end{cases}$$

El error de clasificación $\mathcal{E}_{mis}$ (*mis: missclasification*) de la regla $\mathfrak{x}$ y el factor de coste $\mathbf{c}$ es el error porcentual dado por:

$$\mathcal{E}_{mis} = \left(\frac{1}{n} \sum_{i=1}^{n} c_i \right) \times 100$$

Si $\mathbf{X}_i$ ha sido incorrectamente clasificado, ponemos $c_i = 1$. Un error de clasificación con factores de coste 0 y 1 es el más natural, ya que no se fija en la cantidad de grupos, sino que solo cuenta el número de observaciones mal clasificadas.

Las medidas de evaluación juegan un papel importante cuando la regla se construye en base a una parte de la muestra, para después ser aplicada al conjunto de datos restante. Para $\mathbb{X} = [\mathbf{X}_1\mathbf{X}_2...\mathbf{X}_n]$, sea $\mathbb{X}_0$ formado por $m < n$ de las observaciones originales, y sea $\mathbb{X}_p$ el subconjunto constituido por las $n-m$ observaciones restantes. Primero derivamos una regla $\mathfrak{x}_0$ a partir de $\mathbb{X}_0$ (sin referirnos a $\mathbb{X}_p$) y después predecimos la pertenencia a la clase para los datos de $\mathbb{X}_p$ usando $\mathfrak{x}_0$. Finalmente, calculamos el error de clasificación de la regla $\mathfrak{x}_0$ para las observaciones de $\mathbb{X}_p$. Este proceso es el pilar de la Clasificación Supervisada, fundamentada en muestras de entrenamiento y validación. Hay muchos modos de escoger $\mathbb{X}_0$, pero ahora consideraremos el caso más simple $\mathbb{X}_0$, correspondiente a todas las observaciones menos una.

Definición 1.9. Consideremos el conjunto de datos etiquetado $\mathbb{X} = [\mathbf{X}_1\mathbf{X}_2...\mathbf{X}_n]$ y una regla $\mathfrak{x}$. Para $i \leq n$, sean:

$$\mathbb{X}_{0,(-i)} = [\mathbf{X}_1\mathbf{X}_2...\mathbf{X}_{i-1}\ \ \mathbf{X}_{i+1}\mathbf{X}_n]$$
$$\mathbb{X}_{p,(-i)} = \mathbf{X}_i$$

y llamemos $\mathbb{X}_{0,(-i)}$ la i-ésima muestra de entrenamiento, dejando uno fuera. Sea $\mathfrak{x}_{(-i)}$ la regla construida como $\mathfrak{x}$ pero solo basada en $\mathbb{X}_{0,(-i)}$ y sea $\mathfrak{x}_{(-i)}(\mathbf{X}_i)$ el valor que la regla $\mathfrak{x}_{(-i)}$ asigna a la observación $\mathbf{X}_i$. Un factor de coste $\boldsymbol{k} = [k_{-1}, ..., k_{-n}]$ asociado con las regla $\mathfrak{x}_{(-i)}$ está definido por:

$$k_{-i} = \begin{cases} = 0 & \text{si } \mathbf{X}_i \text{ ha sido correctamente clasificado por } \mathfrak{x}_{(-i)} \\ > 0 & \text{si } \mathbf{X}_i \text{ ha sido incorrectamente clasificado por } \mathfrak{x}_{(-i)} \end{cases}$$

El error dejando uno fuera $\mathcal{E}_{loo}$ (*loo*: *leave-one-out*) basado en las n reglas $\mathfrak{x}_{(-i)}$, con $i \leq n$, los valores asignados $\mathfrak{x}_{(-i)}(\mathbf{X}_i)$ y el factor de coste $\mathbf{k}$, es:

$$\mathcal{E}_{loo} = \left(\frac{1}{n}\sum_{i=1}^{n} k_{-i}\right) \times 100$$

La elección de estos n conjuntos de entrenamiento $\mathbb{X}_{0,(-i)}$, la derivación de de las n reglas $\mathfrak{x}_{(-i)}$ y el cálculo de $\mathcal{E}_{loo}$ recibe el nombre de método dejando uno fuera. Este procedimiento es un caso especial de validación cruzada, denominado validación cruzada de n iteraciones.

Notar que el conjunto $\mathbb{X}_{0,(-i)}$ omite precisamente la observación i-ésima. $\mathbb{X}_{p,(-i)}$ es tenida en cuenta como una nueva observación que necesita ser clasificada por la regla $\mathfrak{x}_{(-i)}$. Este proceso se repite para cada $i \leq n$, de modo que cada observación $\mathbf{X}_i$ se deja fuera exactamente una vez. El error $\mathcal{E}_{loo}$ colecciona contribuciones de aquellas observaciones $\mathbf{X}_i$ para las cuales $\mathfrak{x}_{(-i)}(\mathbf{X}_i)$

difiere del valor de la etiqueta de $\mathbf{X}_i$. Fijarse que las reglas $\mathfrak{r}_{(-i)}$ serán distintas entre ellas y de $\mathfrak{r}$, pero para una buena regla discriminante esta diferencia se hará despreciable a medida que se incrementa el tamaño de la muestra.

Para una evaluación teórica de las reglas, la probabilidad de error es un concepto importante. Introducimos esta noción para problemas binarios (i.e, de dos clases).

Definición 1.10. Sea $\mathbf{X}$ un vector aleatorio perteneciente a una de las dos clases $\mathcal{C}_1$ y $\mathcal{C}_2$, y sea Y la etiqueta de $\mathbf{X}$. Sea $\mathfrak{r}$ una regla discriminante para $\mathbf{X}$. La probabilidad de clasificación incorrecta o probabilidad de error de la regla $\mathfrak{r}$ es $\mathbb{P}\{\mathfrak{r}(\boldsymbol{X}) \neq Y\}$.

Típicamente queremos que la probabilidad de error sea lo más baja posible, por lo que uno está interesado en aquellas reglas que minimizan este error. Esto nos conduce a la siguiente proposición, que proporciona un método para comparar dos reglas discriminantes.

Proposición 1.11. *Sea $\boldsymbol{X}$ un vector aleatorio que pertenece a una de las dos clases $\mathcal{C}_1$ y $\mathcal{C}_2$, y sea Y la etiqueta de $\boldsymbol{X}$. Sea*

$$p(\boldsymbol{X}) = \mathbb{P}\{Y = 1|\boldsymbol{X}\}$$

la probabilidad condicional de que $Y = 1$ dado $\boldsymbol{X}$, y sea $\mathfrak{r}^*$ la regla discriminante definida como:

$$\mathfrak{r}^*(\boldsymbol{X}) = \begin{cases} 1 & si\ p(\boldsymbol{X}) > \frac{1}{2} \\ 0 & si\ p(\boldsymbol{X}) < \frac{1}{2} \end{cases}$$

Si $\mathfrak{r}$ es otra regla discriminante para $\boldsymbol{X}$, decimos que $\mathfrak{r}^*$ es preferible a $\mathfrak{r}$ (o que $\mathfrak{r}^*$ es óptima) si se verifica:

$$\mathbb{P}\{\mathfrak{r}^*(\boldsymbol{X}) \neq Y\} \leq \mathbb{P}\{\mathfrak{r}(\boldsymbol{X}) \neq Y\}$$

1.7. Clasificación mediante las Máquinas de Vector Soporte

En esta sección analizaremos las Máquinas de Vector Soporte o MVS (del inglés: *Support Vector Machines*, abreviado como SVM), una herramienta de clasificación desarrollada por el matemático ruso V. Vapnik [2] en los años 90 y considerada, actualmente, como uno de los mejores métodos de predicción.

Las Máquinas de Vector Soporte son una extensión del Clasificador de Vector Soporte (*Support Vector Classifier*), y éste, a su vez, es una generalización del Clasificador de Margen Maximal (*Maximal Margin Classifier*), el cual, a pesar de ser simple y elegante, no puede ser aplicado a muchos conjuntos de datos debido a su requisito de que las clases sean linealmente separables. Estudiaremos estos 3 conceptos en detalle, subrayando las diferencias entre uno y otro. Para ello, nos basaremos en [7].

1.7.1. Clasificador de Margen Maximal

Supongamos que estamos en la situación de la Figura 1.2 , donde tenemos un conjunto de muestras positivas y negativas que deseamos separar mediante una línea recta. La pregunta es, ¿qué línea recta debemos escoger para obtener una separación óptima de los datos?

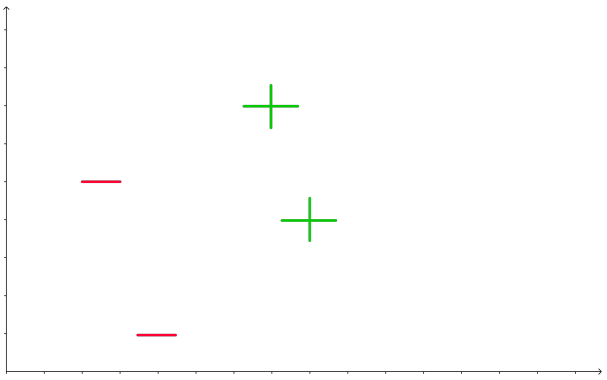

Figura 1.2: Observaciones pertenecientes a dos clases distintas que deseamos separar de forma óptima.

Definamos el margen como la mínima distancia (perpendicular) de cada observación a una recta dada. El Clasificador de Margen Maximal se basa en tomar la línea recta (o hiperplano, válido también para cuando estamos en p dimensiones, $p > 1$) para la cual el margen es mayor. Ahora la cuestión es, ¿qué regla de decisión usa esa frontera de decisión?

Para responder a esto, consideramos un vector $\boldsymbol{w}$, perpendicular a la línea recta y de longitud desconocida, y un vector $\boldsymbol{u}$, como se muestra en la Figura 1.3. Proyectamos $\boldsymbol{u}$ en la dirección de $\boldsymbol{w}$, mediante el producto escalar: $\boldsymbol{w} \cdot \boldsymbol{u}$. Cuánta más grande sea está proyección, más seguros estaremos de que $\boldsymbol{u}$ es una muestra positiva.

Matemáticamente, esto lo podemos expresar como:

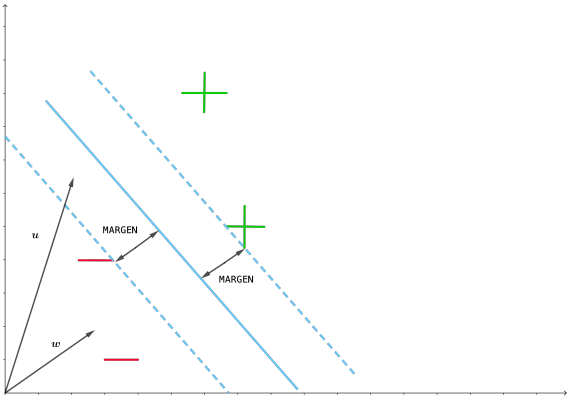

Figura 1.3: Vectores $\boldsymbol{w}, \boldsymbol{u}$ y margen. El hiperplano de margen maximal se muestra como una línea sólida. El margen es la distancia desde la línea sólida hasta cualquiera de las líneas discontinuas.

$$\boldsymbol{w} \cdot \boldsymbol{u} \geq c \Longleftrightarrow [\boldsymbol{w} \cdot \boldsymbol{u} + b \geq 0, \text{ entonces } +] \tag{1.12}$$

siendo c una constante positiva. En la equivalencia hemos definido $b = -c$. Notar que b es desconocido. Esto último nos proporciona la regla de decisión. Ahora bien, como $\boldsymbol{w}$ y $\boldsymbol{u}$ son incógnitas, necesitamos imponer más restricciones para averiguar su valor. Si $\boldsymbol{x}_+$ es una muestra positiva y $\boldsymbol{x}_-$ una muestra negativa, se verifica:

$$\begin{aligned} \boldsymbol{w} \cdot \boldsymbol{x}_+ + b &\geq 1 \\ \boldsymbol{w} \cdot \boldsymbol{x}_- + b &\leq -1 \end{aligned} \tag{1.13}$$

pues $\boldsymbol{w} \cdot \boldsymbol{x} + b \geq 0$ si $\boldsymbol{x}$ está en la línea recta. Para hacer las cosas más sencillas, introducimos una nueva variable y_i tal que: $y_i = 1$ para muestras positivas, e $y_i = -1$ para datos negativos. Se tiene:

$$\begin{aligned} \boldsymbol{w} \cdot \boldsymbol{x}_+ + b \geq 1 &\Longleftrightarrow y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 \\ \boldsymbol{w} \cdot \boldsymbol{x}_- + b \leq -1 &\Longleftrightarrow y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 \end{aligned}$$

Es decir, las dos ecuaciones son iguales, con lo cual basta considerar solo una de ellas. Así:

$$\begin{aligned} y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 &\Longleftrightarrow y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 \geq 0 \\ y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 = 0, &\text{ para } \boldsymbol{x}_i \text{ en los bordes del hiperplano} \end{aligned}$$

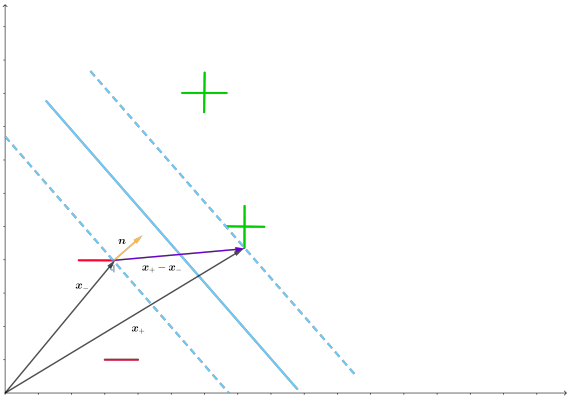

Figura 1.4: Cálculo geométrico de la distancia mínima entre un par de observaciones pertenecientes a clases distintas.

A continuación, calcularemos la distancia entre las líneas discontinuas, por el procedimiento que se ve en la Figura 1.4.

donde $\boldsymbol{n}$ es un vector normal unitario perpendicular a la línea recta. La proyección de $\boldsymbol{n}$ sobre $\boldsymbol{x}_+ - \boldsymbol{x}_-$ nos proporciona la longitud de éste, i.e: la distancia buscada. Ahora bien, ¿qué vector $\boldsymbol{n}$ elegimos? Dado que el vector $\boldsymbol{w}$ definido anteriormente verificaba la perpendicular con respecto a la recta, basta tomar $\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$ para que, además, sea unitario.

Matemáticamente, esto se traduce como:

$$\text{Distancia} = (\boldsymbol{x}_+ - \boldsymbol{x}_-) \cdot \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|}$$

donde en la segunda igualdad hemos tenido en cuenta que, como las observaciones $\boldsymbol{x}_+$ y $\boldsymbol{x}_-$ están en los bordes, se verifica: $y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 = 0$. En particular, para $\boldsymbol{x}_+$, $y_i = 1$, con lo cual $\boldsymbol{w} \cdot \boldsymbol{x}_+ = 1 - b$. De forma similar, para $\boldsymbol{x}_-$ se tiene que $y_i = -1$ y por lo tanto $\boldsymbol{w} \cdot \boldsymbol{x}_- = -1 - b$.

EL hiperplano maximal, bajo las restricciones consideradas se calcula como: máx $\frac{2}{\|\boldsymbol{w}\|}$ y:

$$\text{máx}\, \frac{2}{\|\boldsymbol{w}\|} \iff \text{máx}\, \frac{1}{\|\boldsymbol{w}\|} \iff \text{mín} \|\boldsymbol{w}\| \iff \text{mín}\, \frac{1}{2}\|\boldsymbol{w}\|^2$$

La construcción del Clasificador de Margen Maximal se basa en resolver el siguiente problema de optimización:

$$\begin{aligned} &\text{mín} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{sujeto a: } y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 \geq 0 \end{aligned} \tag{1.14}$$

Para ello, emplearemos los multiplicadores de Lagrange:

$$
\begin{aligned}
L &= \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_i \alpha_i [y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1] \\
\frac{\partial L}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_i \alpha_i y_i \boldsymbol{x}_i = 0 \Longrightarrow \boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i \\
\frac{\partial L}{\partial b} &= -\sum_i \alpha_i y_i = 0 \Longrightarrow \sum_i \alpha_i y_i = 0
\end{aligned}
\tag{1.15}
$$

Substituimos ahora el valor de $\boldsymbol{w}$ en la expresión de L, y obtenemos:

$$
L = \frac{1}{2}\left(\sum_i \alpha_i y_i \boldsymbol{x}_i\right)\left(\sum_j \alpha_j y_j \boldsymbol{x}_j\right) - \left(\sum_i \alpha_i y_i \boldsymbol{x}_i\right)\left(\sum_j \alpha_j y_j \boldsymbol{x}_j\right) - \sum_i \alpha_i y_i b + \sum_i \alpha_i \Longleftrightarrow
$$

$$
L = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j)
$$

Es decir: la optimización de L solo depende del producto escalar entre pares de muestras. Dicho de otro modo: el Clasificador de Margen Maximal depende de un conjunto muy pequeño de observaciones, las cuales son denominadas vectores soporte, ya que "soportan" el hiperplano maximal, en el sentido de que son puntos que si se mueven ligeramente, entonces el hiperplano también se desplaza.

1.7.2. Clasificador de Vector Soporte

No siempre se da el caso en que los datos sean separables por un hiperplano. De hecho, aún en el supuesto de que tal hiperplano existiera, hay situaciones en las que este clasificador no es deseable. Un clasificador basado en un hiperplano separador necesariamente clasificará a la perfección todas las muestras, y esto puede causar una alta sensibilidad a observaciones individuales, produciendo un sobreajuste en los datos.

En esta situación, nos interesa considerar un clasificador basado en un hiperplano que no separe perfectamente las dos clases, para lograr:

- Mayor robustez frente a observaciones individuales.
- Mejor clasificación para la *mayor parte* de muestras.

Esto es, puede merecer la pena clasificar mal unas pocas observaciones para separar mejor los datos restantes.

El Clasificador de Vector Soporte hace exactamente esto: en vez de encontrar el margen más ancho posible para que cada observación esté no solo en el lado correcto del hiperplano, sino también en el lugar apropiado del margen, permitimos mayor flexibilidad admitiendo que algunas observaciones estén en la parte incorrecta del margen o el hiperplano. Las observaciones que verifican esto último son las muestras mal clasificadas por el Clasificador de Vector Soporte.

Matemáticamente, su construcción se basa en resolver el siguiente problema de optimización:

$$\begin{aligned} \text{mín} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{sujeto a: } & y_i \cdot (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 + \xi_i \geq 0 \\ & \xi_i \geq 0, \sum_{i=1}^{n} \xi_i \leq C \end{aligned} \tag{1.16}$$

donde hemos introducido unas nuevas variables no negativas, llamadas variables de holgura ξ_i (del inglés: *slack variables*), las cuales "relajan" las condiciones iniciales del margen, permitiendo que éste sea más flexible, i.e, que algunas observaciones no estén en el lugar correcto.

Siguiendo un razonamiento análogo al realizado para el Clasificador de Margen Maximal 1.13, se tiene que las restricciones en este caso son:

$$\begin{aligned} \boldsymbol{w} \cdot \boldsymbol{x}_+ + b \geq \quad & 1 - \xi_i \\ \boldsymbol{w} \cdot \boldsymbol{x}_- + b \leq & -1 + \xi_i \end{aligned}$$

Asimismo, la variable ξ_i nos indica la posición de la observación $\boldsymbol{x}_i$, relativa al hiperplano y al margen. Si $\xi_i = 0$, $\boldsymbol{x}_i$ está en el lado correcto del margen. Si $\xi_i > 0$, $\boldsymbol{x}_i$ está en el lugar erróneo del margen (decimos que la observación i-ésima ha *violado* el margen). Finalmente, si $\xi_i > 1$, entonces $\boldsymbol{x}_i$ está en el lado incorrecto del hiperplano.

Por su parte, la constante C que aparece en 1.16 es un parámetro de penalización. Su función como cota de la suma de los ξ_i's nos sirve para determinar el número y la severidad de las violaciones del margen y el hiperplano que estamos dispuestos a tolerar. Por ejemplo, si C>0, entonces no más de C observaciones pueden estar en el espacio equivocado del hiperplano. Notar que el caso $C = 0$ se corresponde con el problema de optimización 1.14, pues en esta situación: $\xi_1 = ... = \xi_n = 0$. A medida que C se incrementa nos volvemos más tolerantes a los incumplimientos del margen, resultando en un ensanchamiento de éste.

En la práctica, C se calcula mediante validación cruzada y controla el equilibro sesgo-varianza: cuando C es pequeño, los márgenes son estrechos y rara vez son transgredidos por los datos, lo

cual se traduce en poco sesgo, pero en una alta varianza. Sin embargo, cuando C es grande, el margen es más amplio, derivando en un peor ajuste de las observaciones y en un clasificador potencialmente más sesgado, aunque de menor varianza. La Figura 1.5 ilustra, de forma gráfica, cómo se transforma el margen en función de los valores de C.

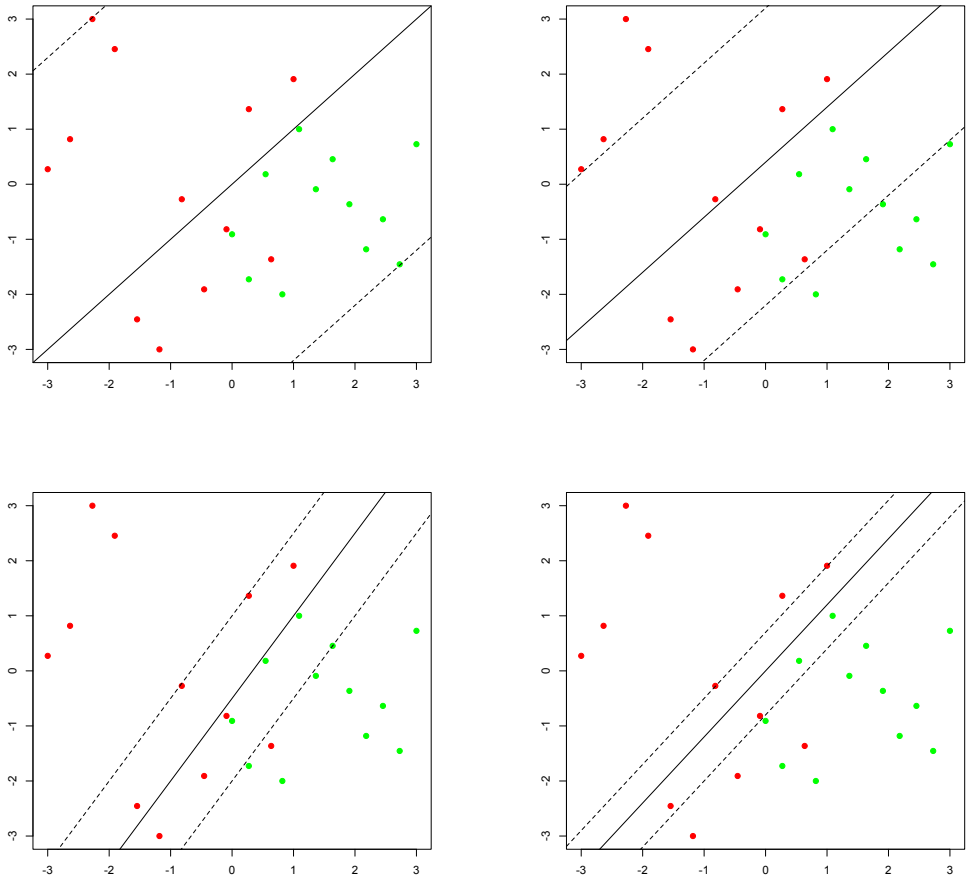

Figura 1.5: Variación del margen para distintos valores de C. El mayor valor de C corresponde al gráfico superior izquierdo, mientras que las restantes figuras se obtienen considerando valores decrecientes del parámetro. Cuando C es grande, hay una alta tolerancia para observaciones que no se encuentran en el lado correcto del margen, por lo que el margen será ancho. A medida que C decrece, la tolerancia para la transgresión del margen también es menor, por lo que el margen se estrecha.

El problema de optimización 1.16 tiene una propiedad muy interesante: solo las muestras que, o bien descansan en el margen, o bien lo vulneran, afectarán al hiperplano, y por lo tanto, al clasificador. Estas observaciones son los vectores soporte y, en consonancia con lo descrito en el párrafo precedente, aumentan cuando C es alto y disminuyen si éste es pequeño. Esta carac-

terística muestra la robustez ante el comportamiento de observaciones alejadas del hiperplano. Además, distingue al Clasificador de Vector Soporte de otros métodos de clasificación, como el LDA. En efecto, la regla de clasificación del LDA 1.5 depende de la media de *todas* las observaciones entre cada clase, así como de la matriz de varianzas-covarianzas entre-grupos empleando *todos* los datos.

1.7.3. Máquinas de Vector Soporte

El Clasificador de Vector Soporte es un enfoque natural para la clasificación en el contexto binario, si la frontera entre las dos clases es lineal. No obstante, en la práctica, muchas veces esto no se verifica y en estas situaciones cualquier clasificador lineal actuará pobremente.

Las Máquinas de Vector Soporte solucionan esta problema aplicando una transformación de los datos originales a un nuevo espacio, de dimensión superior, donde sí se pueden separar de forma lineal. La ampliación del espacio se realiza mediante unas funciones específicas, denominadas kernels. Veamos esto en detalle.

Fijémonos en que la solución del problema de optimización 1.16 y el Clasificador de Vector Soporte lineal solo dependen del producto escalar entre las observaciones:

$$\begin{aligned} L \quad &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \\ h(x) &= b + \sum_{i=1}^{n} \alpha_i y_i (\boldsymbol{x}_i \cdot \boldsymbol{u}) \end{aligned}$$

donde la expresión de h viene de considerar la regla de decisión 1.12 y el valor de $\boldsymbol{w}$ obtenido en 1.15. Notar que $\alpha_i \neq 0$ si y solo si $\boldsymbol{x}_i$ es un vector soporte.

Si ahora consideramos una transformación φ, para hacer la maximización solo necesitaríamos conocer el valor de: $\varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}_j)$, mientras que para la regla de decisión bastaría: $\varphi(\boldsymbol{x}_j) \cdot \varphi(\boldsymbol{u})$. En otras palabras, no necesitamos conocer la transformación φ, sino solo el producto escalar de los vectores transformados, y aquí es donde reside la magia de las Máquinas de Vector Soporte.

Las funciones kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}_j)$ son una generalización del producto escalar, que cuantifica la similaridad entre un par de observaciones.

Algunos kernel populares son:

- El kernel lineal: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$, que mide el parecido entre dos muestras usando la correlación (estándar) de Pearson.

- El kernel polinómico de grado p: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^p$, con $p>1$.
- El kernel radial (en inglés: *radial basis function kernel* o RBF kernel): $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$.

Cuando el Clasificador de Vector Soporte se combina con un kernel no lineal como el polinómico o el radial, el clasificador resultante recibe el nombre de Máquina de Vector Soporte. En este caso, la función de decisión (no lineal) toma la forma:

$$h(x) = b + \sum \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Veamos, por ejemplo, cómo funciona el kernel radial. Si dada una nueva observación $\boldsymbol{x}_i^*$, ésta se encuentra muy alejada de una muestra de entrenamiento $\boldsymbol{x}_i$, i.e, si la distancia euclídea $\|\boldsymbol{x}_i - \boldsymbol{x}_i^*\|^2$ entre ambas es grande, K será pequeño. Esto significa que en la ecuación anterior, $\boldsymbol{x}_i$ no jugará ningún papel en $h(\boldsymbol{x}_i^*)$ o, equivalentemente, en la predicción de la pertenencia de $\boldsymbol{x}_i^*$ a una u otra clase. Se dice entonces que el kernel radial tiene un comportamiento muy local, en el sentido de que solo muestras de entrenamiento cercanas tienen un efecto en la etiqueta de la clase correspondiente a una muestra de validación. La Figura 1.6 es un ejemplo de cómo actúan los kernels radial y polinómico cuando los aplicamos a un conjunto de datos que no se pueden separar de forma lineal.

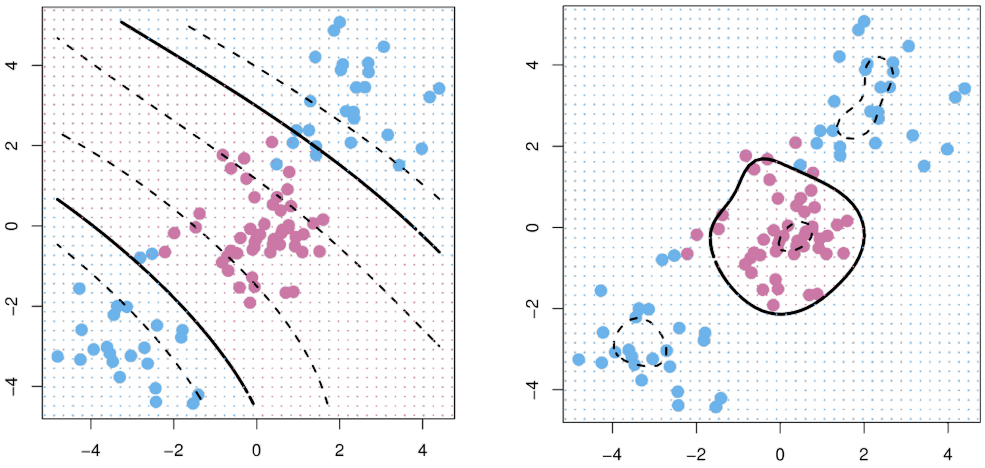

Figura 1.6: Ajuste por MVS de un conjunto de observaciones no linealmente separables, pertenecientes a dos clases distintas (señaladas en rosa y azul). La gráfica de la izquierda corresponde a una MVS con un kernel polinómico de grado 3 y la de la derecha a un kernel radial. Imágenes extraídas de [7]

La ventaja de usar las funciones kernel K es esencialmente computacional, ya que solo precisamos calcular $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, para todos los $\binom{n}{2}$ pares distintos i, j. Esto se puede realizar sin trabajar

explícitamente en el espacio ampliado y es muy importante, pues en numerosas aplicaciones de las MVS el nuevo espacio tiene tantas dimensiones que los cálculos se vuelven intratables.

Por último, es relevante mencionar que el concepto de hiperplano separador en el cual se basan las MVS's no se extiende de forma natural para más de dos clases. No obstante, hay algunos métodos que intentan generalizar este argumento para el caso multiclase, siendo los más conocidos el *one-versus-one* y el *one-versus-all*.

1.8. LDA generalizado

Recordemos algunas de las ventajas del Análisis Discriminante Lineal.

- La regla de decisión solo depende de la media de cada clase. Recordar que por 1.2, una nueva observación es asignada a la clase de media más cercana.
- Las fronteras de decisión son lineales, originando, con ello, reglas de clasificación simples en forma e implementación.
- Actúa como reductor de dimensiones, lo cual permite, entre otras cosas, una mejor visualización de los datos.
- Suele producir los mejores resultados de clasificación, debido a su sencillez y baja varianza.

Sin embargo, la simplicidad característica de esta técnica de clasificación provoca que falle en numerosas situaciones:

- En muchas ocasiones se necesita más de una medida para describir y clasificar los datos.
- A menudo las fronteras de decisión lineales no separan adecuadamente las clases.
- En aquellos casos donde tenemos demasiadas variables correladas, LDA usa un gran número de parámetros, que son estimados con una alta varianza, y su rendimiento es pobre.

Con el fin de solucionar estos problemas, se crearon una serie de técnicas que generalizan el modelo del Análisis Discriminante Lineal ([6]). Las describiremos brevemente.

La primera idea consiste en pensar el problema del LDA como un problema de regresión lineal. Hay muchos métodos que generalizan la regresión lineal para obtener más flexibilidad. Esto, en consecuencia, conduce a formas más flexibles de análisis discriminante, que llamamos Análisis Discriminante Flexible o FDA (del inglés: *Flexible Discriminant Analysis*). En la mayor parte de

los casos los procedimientos consisten en ampliar el conjunto de variables mediante funciones no lineales de los predictores. EL FDA equivale al LDA en el espacio ampliado, el mismo argumento usado en las MVS's. Es útil para modelar relaciones no lineales entre las variables.

En el caso de muchos predictores, la segunda idea radica en ajustar un modelo LDA, pero penalizando los coeficientes, de forma similar a las técnicas de regularización en el contexto de regresión. A este procedimiento se le conoce como Análisis Discriminante Penalizado o PDA (del inglés: *Penalized Discriminant Analysis*).

El clasificador LDA asume que cada clase proviene de una sola distribución gaussiana. Esto, en ocasiones, puede ser muy restrictivo. La tercera idea se fundamenta en modelar cada clase como una mezcla de dos o más subclases con distribuciones normales y con la misma matriz de varianzas-covarianzas. Ello permite fronteras de decisión más complejas, así como una reducción de dimensiones al igual que en el LDA. Esta extensión se la conoce con el nombre de Análisis Discriminante Mixto o MDA (del inglés: *Mixture Discriminant Analysis*).

Capítulo 2

Adaptación al contexto de alta dimensión en "Big Data"

El avance de la ciencia y la tecnología trae consigo conjuntos de datos donde el número de variables es mucho mayor que el tamaño de la muestra. En tales situaciones los clásicos LDA y QDA son inaplicables, pues las matrices de varianzas-covarianzas no poseen inversa. En los últimos años, los estadísticos han dedicado grandes esfuerzos a adaptar estos métodos para dimensiones elevadas. Hay una ingente cantidad de información sobre este tema y multitud de propuestas, pero en este capítulo nos centraremos solo en algunas de ellas, las cuales, por sus características, resultan particularmente interesantes.

2.1. Reglas de clasificación en alta dimensión para LDA

La fórmula para el clasificador LDA, considerando dos clases ($k = 2$) con distribución normal y un vector de predictores p-dimensional $\mathbf{X}$ es:

$$\mathfrak{x}_{LDA}(\mathbf{X}) = \mathbb{I}\left(\mathbf{X} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2)^T \hat{\Sigma}^{-1}(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2) + \log\frac{\hat{\pi}_1}{\hat{\pi}_2} > 0\right) \tag{2.1}$$

donde $\hat{\boldsymbol{\mu}}_\nu, \hat{\Sigma}^{-1}$ y $\hat{\pi}_\nu$ son los estimadores muestrales de los parámetros poblaciones $\boldsymbol{\mu}_\nu, \Sigma^{-1}$ y π_ν, respectivamente, para $\nu = 1, 2$.

Como vimos en el capítulo anterior, el clasificador 2.1 era óptimo, según la regla de Bayes, si el número de predictores (p) estaba fijado y el tamaño muestral (n) tendía a infinito. En caso contrario, cuando la dimensión es elevada, se presentan esencialmente tres problemas: la singularidad de la matriz $\hat{\Sigma}$, la adecuada estimación de $\hat{\boldsymbol{\mu}}_j$ y la regularización simultánea de

Σ y $\boldsymbol{\mu}_j$. Para solventar estas dificultades se han propuesto una serie de clasificadores en un intento por adaptar LDA a dimensiones superiores, que se pueden agrupar, esencialmente, en dos categorías: clasificadores no dispersos y clasificadores dispersos. A continuación introducimos algunos ejemplos de cada tipo, extraídos de [9].

2.1.1. Clasificadores no dispersos

Estos métodos emplean selección de variables mediante técnicas de regularización aplicadas a las medias o la matriz de varianzas-covarianzas. La denominación de "no disperso" hace referencia a que estos clasificadores no necesitan asumir que las matrices sean huecas, o equivalentemente, que tengan un gran número de entradas nulas.

2.1.1.1. Independence Rule (IR)

Esta técnica de clasificación fue creada para solventar el problema de la singularidad de la matriz $\hat{\Sigma}$. IR sustituye la matriz anterior por $\hat{\mathbf{D}} = diag(\hat{\Sigma})$, que es un estimador definido positivo y, por lo tanto, siempre invertible. El hecho de que $\hat{\mathbf{D}}$ sea diagonal equivale a decir que las variables, en cada grupo, son independientes. Se ha comprobado que IR puede proporcionar mejores resultados que una regla que modele todas las posibles correlaciones entre los predictores.

Para definir IR, consideremos el siguiente espacio de parámetros:

$$\Gamma(c, \boldsymbol{\nu}, \mathbf{B}) = \left\{(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) : (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \Sigma^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \geq c^2, \nu_1 \leq \lambda_{min}(\Sigma) \leq \lambda_{max}(\Sigma) \leq \nu_2, \boldsymbol{\mu}_i \in B\right\}$$

donde $\mathbf{B} = \mathbf{B}_{a,d} = \left\{\boldsymbol{\mu} \in \mathcal{L}_2 : \Sigma_{j=1}^{p} a_j \boldsymbol{\mu}_j^2 \leq d^2\right\}$, $\boldsymbol{\nu} = (\nu_1, \nu_2)$ y $\lambda_{min}(\Sigma)$ y $\lambda_{max}(\Sigma)$ son el menor y mayor autovalor de Σ, respectivamente.

Si estimamos $\boldsymbol{\mu}_{ij}$ con:

$$\hat{\boldsymbol{\mu}}_{ij}^{IR} = (1 - r_{jn}) + \hat{\boldsymbol{\mu}}_{ij}$$

para algún r_{jn} adecuadamente escogido, el clasificador IR adopta la forma:

$$\mathfrak{r}_{IR}(\mathbf{X}) = \mathbb{I}\left(\mathbf{X} - \frac{1}{2}(\boldsymbol{\mu}_1^{IR} + \hat{\boldsymbol{\mu}}_2^{IR})^T \hat{\mathbf{D}}^{-1} (\hat{\boldsymbol{\mu}}_1^{IR} - \hat{\boldsymbol{\mu}}_2^{IR}) + \log \frac{n_1}{n_2} > 0\right) \tag{2.2}$$

2.1.1.2. Features Annealed Independence Rule (FAIR)

Se puede probar que IR pierde sus buenas propiedades si la dimensión es demasiado alta. No obstante, si esta misma regla es aplicada a unas variables seleccionadas, la técnica de clasificación resultante FAIR, propuesta por Fan y Fan ([4]), soluciona los problemas de interpretabilidad y acumulación de ruido debido al exceso de variables en el contexto multidimensional, lo cual conlleva estimar un gran número de parámetros.

Con objeto de seleccionar únicamente aquellos predictores que recogen la mayor variabilidad de la muestra se emplea el test t. La elección del número óptimo de variables o, equivalentemente, el valor umbral de los estadísticos t, son propuestos basados en una cota del error de clasificación.

Fijándonos en 2.2 y teniendo en cuenta que $\hat{\mathbf{D}}^{-1}$ es diagonal, vemos que las únicas variables que aportan algo a la regla de decisión son aquellas cuyas medias son diferentes entre clases. Por ello, el estadístico empleado para seleccionar las m componentes más significativas es el estadístico t, definido para el predictor j como:

$$t_j = \frac{\hat{\boldsymbol{\mu}}_{1j} - \hat{\boldsymbol{\mu}}_{2j}}{\sqrt{\frac{s_{1j}^2}{n_1} + \frac{s_{2j}^2}{n_2}}}, \qquad j = 1, ...p$$

donde $s_{\nu j}^2$ es la varianza muestral de X_j en la clase ν. En [4] se pueden ver las condiciones bajo las cuales el estadístico anterior selecciona todas las variables importantes con probabilidad tendiendo a 1.

La regla FAIR selecciona las variables en el conjunto $\hat{S} = \{j : |t_j| \text{ está entre los } s_n \text{ más grandes}\}$

En la práctica, se ordenan los $|t_j|$ por orden decreciente y se fija:

$$s_n = \arg\max_s \frac{1}{\hat{\lambda}_{max}^s} \frac{n\left[\sum_{j=1}^s t_j^2 + \frac{s(n_1-n_2)}{n}\right]^2}{sn_1n_2 + n_1n_2\sum_{j=1}^s t_j^2}$$

donde $\hat{\lambda}_{max}^s$ es el mayor autovalor del bloque superior izquierdo, de dimensión $s \times s$, de la matriz de correlación muestral $\hat{\mathbf{D}}^{-1/2}\hat{\Sigma}\,\hat{\mathbf{D}}^{-1/2}$. Sirve para minimizar una cota superior del error de clasificación.

FAIR clasifica una observación de acuerdo a la siguiente regla de decisión:

$$\mathfrak{r}_{FAIR}(\mathbf{X}) = \mathbb{I}\left(\left(\mathbf{X}_{\hat{S}} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_{1\hat{S}} + \hat{\boldsymbol{\mu}}_{2\hat{S}})\right)^T \times \hat{\mathbf{D}}_{\hat{S}\hat{S}}^{-1}(\hat{\boldsymbol{\mu}}_{1\hat{S}} - \hat{\boldsymbol{\mu}}_{2\hat{S}}) + \log\frac{n_1}{n_2} > 0\right) \tag{2.3}$$

2.1.1.3. Nearest Shrunken Centroids Classifier (PAM)

Este es un célebre clasificador en el ámbito de la genética (*microarray data*) y por ello también se le conoce con el nombre de PAM (del inglés *Prediction Analysis for Microarrays*). Es una modificación del Nearest Centroids (Prototype) Classifier, el cual asigna una nueva observación a la clase con la distancia más pequeña entre la observación y el centroide. Fue diseñado por Hastie, Tibshirani et al. ([14]). Como se explicita en el artículo, la contracción de los prototipos permite obtener un clasificador más eficiente y con mayor poder de predicción que otros métodos existentes, pues reduce el exceso de ruido. Aunque en [14] este clasificador se emplea para la diagnosis objetiva de una muestra en el contexto de la clasificación cancerígena, el procedimiento es general y puede ser usado en muchos otros problemas de clasificación. Además, PAM funciona especialmente bien cuando hay más de dos clases.

Al igual que FAIR, PAM aproxima Σ mediante una matriz diagonal y se hace una selección de variables, pero los estimadores empleados son distintos. En efecto, en este caso:

$$\tilde{\Sigma} = \hat{\mathbf{D}} + s_0\mathbf{I}$$

para alguna constante pequeña $s_0 > 0$, donde $\mathbf{I}$ es la matriz identidad, y:

$$t^*_{\nu j} = \frac{\hat{\boldsymbol{\mu}}_{\nu j} - \hat{\bar{\boldsymbol{\mu}}}_j}{m_\nu(s_j + s_0)}, \qquad \nu = 1, 2,\ j = 1, ...p$$

donde $\bar{\boldsymbol{\mu}}_j = \frac{1}{n}\sum_{i=1}^n X_j^i, m_\nu = \sqrt{(1/n_\nu) + (1/n)}$ y $s_j^2 = \hat{D}_{jj}$

PAM contrae t_j a cero (empleando un valor umbral o *soft thresholding*), proporcionando la variable t'_{kj} y los centroides reducidos: $\hat{\boldsymbol{\mu}}' = \hat{\bar{\boldsymbol{\mu}}}_j + m_k(s_j + s_0)t'_{kj}$. De esta forma se introduce selección de variables en el método. La regla de decisión resulta en:

$$\mathfrak{r}_{PAM}(\mathbf{X}) = \arg\max_\nu (\mathbf{X} - \hat{\boldsymbol{\mu}}'_\nu)^T \tilde{\Sigma}^{-1}(\mathbf{X} - \hat{\boldsymbol{\mu}}'_\nu) - 2\hat{\pi}_\nu$$

2.1.2. Clasificadores dispersos

Los clasificadores mostrados anteriormente tienen sus desventajas. Por ejemplo, intentar regularizar individualmente Σ y μ_j puede conducir a error, tanto en términos de selección de variables como a efectos de clasificación. De hecho, se pueden ver ejemplos donde la selección de variables aplicada a IR escoge un conjunto erróneo de predictores. Además, aunque en FAIR no se requiere ninguna condición de dispersión sobre Σ, este clasificador no es asintóticamente óptimo, pues ignora las correlaciones entre las componentes de $\mathbf{X}$.

Esta sección estará dedicada a presentar algunos ejemplos de clasificadores que asumen la dispersión de $\beta^{Bayes} = \Sigma^{-1}(\boldsymbol{\mu_1} - \boldsymbol{\mu_2})$, i.e, asumen que el número de elementos no nulos en el vector β^{Bayes} es mucho menor que p. Estos métodos emplean, por tanto, la idea de penalización en el contexto de regresión para regularizar la dirección discriminante estimada, donde se forzaba a que los coeficientes de regresión tendiesen a cero.

2.1.2.1. Conceptos y definiciones previas

Consideraremos el problema de clasificación binario y un vector p-dimensional $\mathbf{X}$, perteneciente a la clase ν si $\mathbf{X} \sim \mathcal{N}_p(\boldsymbol{\mu}_\nu, \Sigma)$ para $\nu = 1, 2$, donde $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$ y Σ es definida positiva.

El índice (tasa) de clasificación incorrecta es la media de las probabilidades de cometer dos tipos de errores: clasificar $\mathbf{X}$ a la clase 1, si $\mathbf{X} \sim \mathcal{N}_p(\boldsymbol{\mu}_2, \Sigma)$ y clasificar $\mathbf{X}$ a la clase 2, cuando $\mathbf{X} \sim \mathcal{N}_p(\boldsymbol{\mu}_1, \Sigma)$.

Si $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ y Σ son conocidas, entonces la regla de clasificación óptima, i.e, aquella con el menor error de clasificación, clasifica $\mathbf{X}$ a la clase 1 si y solo si $\boldsymbol{\delta}'\Sigma^{-1}(x - \bar{\boldsymbol{\mu}}) \geq 0$, donde $\bar{\boldsymbol{\mu}} = (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2, \boldsymbol{\delta} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ y $\boldsymbol{\delta}'$ denota el vector traspuesto del vector $\boldsymbol{\delta}$. Notar que estamos ante la regla de Bayes cuando las probabilidades a priori son iguales. Denotemos por R_{OPT} la tasa de clasificación incorrecta de la regla óptima. Se tiene:

$$R_{OPT} = \Phi(-\Delta_p/2), \qquad \Delta_p = \sqrt{\boldsymbol{\delta}'\Sigma^{-1}\boldsymbol{\delta}} \tag{2.4}$$

donde Φ es la función de distribución de una normal estándar.

Como en la práctica $\boldsymbol{\mu}_\nu$ y Σ son desconocidos, precisamos de una muestra de entrenamiento $\mathbf{X} = \{\mathbf{x}_{\nu i}, i = 1, ..., n_\nu,\ \nu = 1, 2\}$, donde n_ν es el tamaño muestral de la clase ν y $\mathbf{x}_{\nu i} \sim \mathcal{N}_p(\boldsymbol{\mu}_\nu, \Sigma)$ son independientes. Para evaluar el buen funcionamiento de una regla de clasificación T basada en una muestra de entrenamiento, se utiliza la tasa de clasificación incorrecta condicionada $R_T(\mathbf{X})$, definida como la media de las probabilidades condicionales de cometer los dos tipos de clasifiación érronea antes mencionados, donde las probabilidades condicionadas se miden respecto a $\mathbf{x}$, dada la muestra de entrenamiento $\mathbf{X}$. El comportamiento asintótico de T es equivalente a analizar cómo actúa $R_T(\mathbf{X})$ cuando $n \to \infty$.

Definición 2.1. Sea T una regla de clasificación con probabilidad de error condicionada $R_T(\mathbf{X})$, dada la muestra de entrenamiento $\mathbf{X}$.

I. T es asintóticamente óptima si $R_T(\mathbf{X})/R_{OPT} \to_P 1$

II. T es asintóticamente sub-óptima si $R_T(\mathbf{X}) - R_{OPT} \to_P 0$

III. T es asintóticamente peor si $R_T(\mathbf{X}) \to_P 1/2$

Si $\lim_{n\to\infty} R_{OPT}$ [$\iff \Delta_p$ en 2.4 está acotado], entonces (I) y (II) son equivalentes. El apartado (III) viene del hecho de que $1/2$ es la probabilidad de clasificación incorrecta de una elección al azar.

Nos centraremos en reglas de clasificación de la forma:

$$\text{Asignar } \mathbf{x} \text{ a la clase } 1 \iff \hat{\boldsymbol{\delta}}' \hat{\Sigma}^{-1}(\mathbf{x} - \hat{\bar{\boldsymbol{\mu}}}) \geq 0 \tag{2.5}$$

donde $\hat{\boldsymbol{\delta}}, \hat{\bar{\boldsymbol{\mu}}}$ y $\hat{\Sigma}^{-1}$ son los estimadores de $\boldsymbol{\delta}, \bar{\boldsymbol{\mu}}$ y Σ^{-1} usando la muestra de entrenamiento $\mathbf{X}$.

2.1.2.2. Linear Programming Discriminant (LPD)

En contraste con otros métodos precedentes basados en la estimación individual de la matriz Σ^{-1} o en el vector diferencia de medias $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, LPD obtiene un clasificador más simple y eficiente a nivel computacional estimando directamente el producto $\beta^{Bayes} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$.

La obtención de la regla proviene del siguiente hecho: $\Sigma\beta^{Bayes} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. Por tanto, el estimador de β^{Bayes} es:

$$\hat{\beta}^{LPD} = \arg\min_{\beta} \|\beta\|_1, \quad \text{tal que} \quad \|\hat{\Sigma}\beta - (\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)\|_\infty \leq \lambda$$

El resultante $\hat{\beta}^{LPD}$ normalmente es disperso y se puede implementar fácilmente usando programación lineal. La dispersión en el producto $\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ es una condición más débil y flexible que imponer la dispersión tanto en Σ^{-1} como en $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. En particular, LPD no requiere que Σ^{-1} sea dispersa. Esto último es una gran ventaja, pues estimar la inversa de una matriz hueca con una elevada dimensión es complejo y requiere de una ingente cantidad de cálculos.

Asumiendo mismas probabilidades a priori, $\pi_1 = \pi_2$, LPD clasifica una nueva observación a la clase 1 si:

$$\left(\mathbf{X} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2)\right)^T \hat{\beta}^{LPD} > 0 \tag{2.6}$$

Se tiene que, bajo ciertas condiciones de regularidad, la probabilidad de error de LPD tenderá a la probabilidad de error de Bayes (i.e, es asintóticamente óptima).

2.1.2.3. Sparse Linear Discriminant Analysis (SLDA)

Como en la situación $p > n$, LDA puede ser asintóticamente peor incluso aunque Σ sea conocida (en cuyo caso la estimación sería exacta), para obtener un LDA óptimo es necesario imponer condiciones de dispersión en Σ y δ cuando ambos son desconocidos. La regla SLDA presentada en el artículo [12] se construye basándose en estimadores de umbral para las medias y la matriz de varianzas-covarianzas, al igual que en FAIR y PAM, pero, a diferencia de éstos, se permitirá que el número de estimadores no nulos sea mucho mayor que n. Esto permitirá asegurar la optimalidad asintótica de la regla de clasificación, i.e, $R_{SLDA}(\mathbf{X})$ convergerá en probabilidad al mismo límite que R_{OPT} y si $R_{OPT} \to 0$, $R_{SLDA}(\mathbf{X})$ y R_{OPT} tendrán, además, la misma velocidad de convergencia.

- *Condición de dispersión para* Σ

  Consideramos:

$$C_{h,p} = \max_{j \leq p} \sum_{l=1}^{p} |\sigma_{jl}|^h \tag{2.7}$$

  donde σ_{jl} es el elemento que ocupa la posición (j,l) en la matriz Σ, y h es una constante que no depende de p tal que $0 \leq h < 1$. Si h=0, $C_{0,p}$ es el mayor número de elementos no nulos de las filas de Σ . Por tanto, $C_{0,p} << p$ implica que hay muchos elementos en Σ iguales a cero. Si $C_{h,p} << p$ para una constante $h \in (0,1)$, entonces Σ es dispersa, en el sentido de que gran parte de sus entradas son muy pequeñas.

  El estimador resultante de Σ, $\tilde{\Sigma}$, se consigue teniendo en cuenta 2.7 e imponiendo el valor umbral $t_n = M_1\sqrt{\log p}/\sqrt{n}$ a los elementos de S situados fuera de la diagonal, donde S es la matriz de covarianzas muestral (i.e, el estimador de Σ obtenido por máxima verosimilitud) y M_1 es una constante positiva.

- *Condición de dispersión para* δ

  Consideramos:

$$D_{g,p} = \sum_{j=1}^{p} \delta_j^{2g} \tag{2.8}$$

  donde δ_j es la j-ésima componente de δ y g es una constante independiente de la dimensión p tal que $0 \leq g < 1$. Como antes, si $D_{g,p} << p$ para $g \in [0,1)$, entonces δ es disperso.

  El estimador disperso de δ, $\tilde{\delta}$, se obtiene imponiendo el valor umbral $a_n = M_2 \left(\log p/n\right)^{\alpha}$ a las componentes de $\hat{\delta}$, donde $M_2 > 0$ y $\alpha \in (0, 1/2)$.

La regla SLDA se obtiene al sustituir $\hat{\boldsymbol{\delta}}$ y $\hat{\Sigma}^{-1}$ en 2.5 por $\tilde{\boldsymbol{\delta}}$ y $\tilde{\Sigma}^{-1}$, donde las constantes M_1 y M_2 se obtienen por validación cruzada.

$$\text{Asignar } \mathbf{x} \text{ a la clase } 1 \Longleftrightarrow \tilde{\boldsymbol{\delta}}^{'}\tilde{\Sigma}^{-1}(\mathbf{x} - \hat{\boldsymbol{\mu}}) \geq 0 \tag{2.9}$$

El Teorema 3 del artículo garantiza la optimalidad asintótica del SLDA bajo ciertas hipótesis en la velocidad de divergencia de $p, C_{h,p}, D_{g,p}$ y Δ_p^2.

2.1.2.4. Direct Sparse Discriminant Analysis (DSDA)

Este clasificador considera LDA como un problema de regresión lineal simple, con lo cual es particularmente eficiente en términos de computación. Definimos:

$$(\hat{\beta}_0^{ols}, \hat{\beta}_0^{ols}) = \arg \min_{(\beta_0,\beta)} \sum_{i=1}^{n} \left(Y_i - \beta_0 - \mathbf{X}_i^T\beta\right)^2 \tag{2.10}$$

donde Y_i es la etiqueta de clase, pero tratada como una variable continua. Por [6], β^{ols} y $\hat{\beta}^{Bayes}$ comparten la misma dirección o, equivalentemente, son proporcionales. Así, DSDA obtiene un estimador disperso como:

$$\hat{\beta}^{DSDA} = \arg \min_{\beta} \sum_{i=1}^{n} \left(Y_i - \beta_0 - \mathbf{X}_i^T\beta\right)^2 + P_\lambda(\beta) \tag{2.11}$$

donde P_λ es una función de penalización en el parámetro λ. De esta forma, la ecuación 2.11 tiene la misma forma que el estimador por mínimos cuadrados de la regresión.

Otra característica de este método es que puede tratar problemas con distintas probabilidades a priori. Considerando $\hat{\beta}^{DSDA}$, DSDA calcula:

$$\hat{\beta}_0^{DSDA} = -\frac{1}{2}\left(\hat{\boldsymbol{\mu}}_1 + \hat{\boldsymbol{\mu}}_2\right)^T \hat{\beta}^{DSDA} + \log\left(\frac{n_1}{n_2}\right) \frac{(\hat{\beta}^{DSDA})^T\hat{\Sigma}\hat{\beta}^{DSDA}}{(\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)^T\hat{\beta}^{DSDA}}$$

Por tanto, DSDA clasifica una nueva observación a la clase 1 si:

$$\mathbf{X}^T\hat{\beta}^{DSDA} + \hat{\beta}_0^{DSDA} > 0 \tag{2.12}$$

2.2. Reglas de clasificación en alta dimensión para QDA

Los clasificadores LDA no proporcionan buenos resultados si las matrices de covarianzas son distintas entre poblaciones, pues el LDA ignora estas diferencias. A día de hoy existen muchos menos clasificadores cuadráticos en altas dimensiones que lineales. Esto se debe a la dificultad intrínseca que conlleva estimar la regla de clasificación cuadrática y, en especial, las matrices de varianzas-covarianzas. Los métodos existentes, creados para reducir tal complejidad, se pueden agrupar en tres categorías:

- Clásico QDA después de una reducción de dimensiones.
- QDA con estimadores de $\boldsymbol{\mu}_\nu$ y Σ_ν no dispersos.
- QDA con estimadores de $\boldsymbol{\mu}_\nu$ y Σ_ν dispersos.

Los clasificadores dispersos suelen gozar de buenas propiedades asintóticas, pero requieren de complicadas asunciones y poseen un coste computacional elevado. Por el contrario, aquellos métodos con estimadores no dispersos tienen mayores tasas de error, pero son más fáciles de implementar a nivel algorítmico.

En esta sección nos centraremos en el problema de clasificación binario, considerando, además, mismas probabilidades a priori e igual probabilidad de clasificación incorrecta. Los métodos presentados están sacados de [11].

2.2.1. Conceptos previos

La regla cuadrática QDA asigna una nueva observación $\mathbf{x}$ a la clase 1 si:

$$\mathfrak{r}_{QDA}(\mathbf{X}) = \mathbb{I}\left((\mathbf{X}-\boldsymbol{\mu}_1)^T\Sigma_1^{-1}(\mathbf{X}-\boldsymbol{\mu}_1) - (\mathbf{X}-\boldsymbol{\mu}_2)^T\Sigma_2^{-1}(\mathbf{X}-\boldsymbol{\mu}_2) + \log\frac{|\Sigma_1|}{|\Sigma_2|} > 0\right) \tag{2.13}$$

donde $|\Sigma|$ denota el determinante de la matriz Σ.

El enfoque tradicional para implementar 2.13 consiste en sustituir las $\boldsymbol{\mu}_\nu$ y Σ_ν por sus correspondientes análogos muestrales. Sin embargo, cuando la dimensión supera al número de datos, los estimadores para las matrices de covarianzas no poseen inversa y, por tanto, el procedimiento anterior es inaplicable. A continuación presentaremos tres ejemplos de clasificadores cuadráticos, cada uno perteneciente a un grupo distinto de los mencionados anteriormente, que tratan de solucionar este problema de la singularidad.

2.2.2. Ridge-forward QDA (RFQD)

Este método reduce la dimensión de los predictores para que sea menor que el tamaño muestral. De esta forma, se pueden usar los estimadores muestrales para las medias y las matrices de varianzas-covarianzas y el clásico QDA funciona bien.

Como se describe en el artículo [11], RFQD emplea una versión modificada del criterio de información de Bayes junto con un método forward para la selección de variables. Es decir, RFQD elabora la regla de clasificación basándose en las herramientas tradicionales para la construcción de modelos en el contexto de la regresión. Así, una nueva observación es asignada a la clase 1 empleando únicamente las variables seleccionadas. Este clasificador está justificado teóricamente si las muestras de entrenamiento son independientes.

2.2.3. High-dimensional Quadratic Classifiers in Non-Sparse settings

El clasificador QDA clásico no siempre produce buenos resultados aún en el caso de que las matrices de varianzas-covarianzas sean conocidas y las poblaciones sigan una distribución normal. El método presentado a continuación aporta información sobre la heterogeneidad entre clases expandiendo tanto el vector diferencia de medias como las matrices de covarianzas, sin necesidad de asumir dispersión en $\boldsymbol{\mu}_\nu$, Σ_ν o Σ_{12}. Se basa en sustituir Σ_ν^{-1} en 2.13 por $\mathbf{A}_\nu$, donde $\mathbf{A}_\nu$ es una matriz definida positiva verificando:

$$tr[\Sigma_\nu(\mathbf{A}_{\nu'} - \mathbf{A}_\nu)] = tr(\mathbf{A}_\nu^{-1}\mathbf{A}_{\nu'}) - p, \quad \text{para } \nu \neq \nu', \ \ \nu, \nu' \in \{1, 2\}$$

La nueva regla de clasificación incluye, además, el término $tr(S_\nu \mathbf{A}_\nu)/n_\nu$, cuya función es reducir el sesgo en cada clase.

Por tanto, una nueva observación $\mathbf{x}$ es asignada a la clase 1 si $W_1(\mathbf{A}_1) - W_2(\mathbf{A}_2) > 0$, donde:

$$W_\nu(\mathbf{A}_\nu) = (\mathbf{x} - \mu_\nu)^T \mathbf{A}_\nu (\mathbf{x} - \mu_\nu) - tr(S_\nu \mathbf{A}_\nu)/n_\nu - log|\mathbf{A}_\nu|, \quad \nu = 1, 2$$

En particular, los autores consideran cuatro tipos de matrices $\mathbf{A}_i$:

$$\textbf{(I)}\ \mathbf{A}_\nu = \mathbf{I}_p; \ \ \textbf{(II)}\ \mathbf{A}_\nu = \frac{p}{tr(\Sigma_\nu)}\mathbf{I}_p; \ \ \textbf{(III)}\ \mathbf{A}_\nu = \Sigma_{\nu(d)}^{-1}; \ \ \textbf{(IV)}\ \mathbf{A}_\nu = \Sigma_\nu^{-1}$$

siendo $\Sigma_{\nu(d)}^{-1}$ la versión diagonalizada de Σ_ν.

Ignorando el término corrector del sesgo, se tiene que el caso (**I**) conduce al clasificador LDA y a un clasificador QDA simplificado, donde no hay necesidad de estimar las matrices de

covarianzas; (**II**) es un caso particular del análisis discriminante regularizado (RDA); (**III**) es el QDA diagonalizado, el cual disminuye la cantidad de parámetros desconocidos en las matrices de varianzas-covarianzas (en concreto, se pasa de un orden de p^2 a p) y por último el caso (**IV**) es el QDA clásico, aplicable únicamente en aquellas situaciones donde el tamaño muestral es mayor que la dimensión. Además, bajo ciertas hipótesis, la tasa de error de la regla de clasificación obtenida con cualquiera de estas cuatro matrices converge, en probabilidad, a cero.

2.2.4. Sparse QDA for high-dimensional data (SQDA)

Cuando p es mucho más pequeño que n, QDA tiene la menor tasa de clasificación incorrecta en términos asintóticos. El SQDA mostrado aquí busca lograr este mismo objetivo para aquellas situaciones donde la dimensión es mucho mayor que el tamaño muestral. Para ello, impone condiciones de dispersión en las medias y las matrices de covarianzas, que supone desconocidas. Veamos el procedimiento para obtener esta regla discriminante.

La función $\mathfrak{x}_{QDA}$ en 2.13 puede ser reescrita como:

$$\mathfrak{x}_{QDA}(\mathbf{x}) = \mathbb{I}\left((\mathbf{x}-\boldsymbol{\mu}_1)^T\nabla(\mathbf{x}-\boldsymbol{\mu}_1) - 2\delta'\Sigma_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \delta\Sigma_2^{-1}\delta + \log\frac{|\Sigma_1|}{|\Sigma_2|}\right) > 0 \qquad (2.14)$$

donde $\boldsymbol{\delta} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ y $\nabla = \Sigma_1^{-1} - \Sigma_2^{-1}$. Las condiciones de dispersión se impondrán en $\boldsymbol{\delta}, \Sigma_1, \Sigma_2$ y $\Delta = \Sigma_1 - \Sigma_2$.

- *Condición de dispersión y valor umbral para estimar δ*

  Definimos $d_p = \sum_{j=1}^{p}|\delta_j|^{2g}$, donde δ_j es la j-ésima componente de $\boldsymbol{\delta}$ y g es una constante en [0,1). La dispersión de δ está garantizada si d_p diverge a ∞ a un ritmo mucho más lento que p, cuando tanto p y n tienden a infinito. Un estimador disperso de $\boldsymbol{\delta}$ se obtiene imponiendo la siguiente cota o valor umbral a los estimadores de máxima verosimilitud de $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$: $t_n = M_0(n^{-1}\log p)^{\alpha}$, para alguna constante $\alpha \in (0, 1/2)$ y $M_0 > 0$.

- *Condición de dispersión y valor umbral para estimar Σ_i y Δ*

  Definimos $c_p = \text{máx}_{\nu=1,2}\,\text{máx}_{i=1,2,\ldots,p}\sum_{j=1}^{p}|\sigma_{\nu ij}|^h$, donde σ_{kij} es la entrada (i,j) de Σ_ν y h es una constante en [0,1). Para asegurar que gran parte de las componentes de Σ_ν sean nulas, necesitamos que c_p sea mucho más pequeña que p. De esta forma, Δ es disperso. Sin embargo, para garantizar la dispersión de las matrices Σ_1 y Σ_2 se necesitan, en este caso, dos valores umbral, determinados por: $t_{1n} = M_1(n^{-1}\log p)^{1/2}$ y $t_{2n} = M_2(n^{-1}\log p)^{1/2}$, para ciertas constantes M_1 y M_2.

La regla SQDA se obtiene entonces sustituyendo $\boldsymbol{\delta}, \Sigma_\nu$ y Δ en 2.14 por sus correspondientes estimadores dispersos. Los parámetros de regularización M_0, M_1 y M_2 son elegidos mediante validación cruzada para minimizar la probabilidad de clasificación incorrecta. Asumiendo ciertas hipótesis, se puede probar que la tasa de error del clasificador SQDA es asintóticamente óptima (i.e, converge a la correspondiente tasa de error de la regla de Bayes).

Capítulo 3

Ilustración sobre datos simulados y reales

En este capítulo ilustraremos, a través del software R, algunas de las técnicas presentadas para un conjunto de datos simulado y dos bases de datos reales. En ambos casos nos centraremos en el problema de clasificación binario, tomando como referencia el artículo [10].

3.1. Datos simulados

Emplear datos simulados tiene varias ventajas. En primer lugar, debido a la escasez de muestras en conjuntos de datos reales, es difícil detectar diferencias significativas y evaluar la relevancia de los clasificadores. En segundo lugar, la falta de información acerca de cómo se generan los datos en un contexto real y la distribución de probabilidad subyacente hace complicado extraer conclusiones. Por último, puesto que conocemos el verdadero comportamiento del modelo en los datos simulados, podemos comprobar cuan bien funciona, lo cual es importante para hacernos una idea de su fiabilidad cuando lo aplicamos a datos reales.

Guiándonos por [10], construiremos varios modelos de simulación diferentes. Para ello, generamos aleatoriamente n muestras, de forma que la probabilidad *a priori* de que una observación pertenezca a una de las dos clases sea la misma, i.e: $\pi_1 = \pi_2$. Esto significa que ambas clases tienen el mismo número de observaciones. Condicionado a la etiqueta de clase ($\nu = 1, 2$), generamos el vector predictor $\mathbf{X}$ de dimensión p de una distribución normal multivariante con vector de medias μ_ν y matriz de varianzas-covarianzas Σ. Sin pérdida de generalidad, tomamos $\mu_1 = 0$ y $\mu_2 = \Sigma\beta^{\text{Bayes}}$ (consideramos ahora $\beta^{\text{Bayes}} = \Sigma^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})$).

Modelo	n	p	Σ	$\beta^{\text{Bayes}} = \Sigma^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})$
1	100	400	$\Sigma_{ij} = 0{,}5^{\lvert i-j \rvert}$	$0{,}556(3, 1{,}5, 0, 0, 2, 0_{p-5})^T$
2	100	400	$\Sigma_{ij} = 0{,}5^{\lvert i-j \rvert}$	$0{,}582(3, 2{,}5, -2{,}8, 0_{p-3})^T$
3	400	800	$\Sigma_{jj} = 1, \Sigma_{ij} = 0{,}5, i \neq j$	$0{,}395(3, 1{,}7, -2{,}2, -2{,}1, 2{,}55, 0_{p-5})^T$
4	400	800	$\Sigma_{jj} = 1, \Sigma_{ij} = 0{,}5, i \neq j$	$0{,}551(3, 1{,}7, -2{,}2, -2{,}1, 2{,}55, (p-5)^{-1}\mathrm{I}_{p-5})^T$
5	400	800	$\Sigma_{jj} = 1, \Sigma_{ij} = 0{,}5, i \neq j$	$0{,}362(3, 1{,}7, -2{,}2, -2{,}1, 2{,}55, (p-5)^{-1}\mathrm{I}_{p-5})^T$

Cuadro 3.1: Valores de los parámetros para distintos modelos de simulación, extraídos de [10].

Las elecciones para n, p, Σ y β^{Bayes} se recogen en la tabla 3.1.

Los modelos 1-3 son todos dispersos, pero poseen una estructura diferente en las medias y la matriz de covarianzas. Los modelos 4 y 5 son prácticamente dispersos en el sentido de que no dependen de todas las variables y se pueden aproximar bien por funciones discriminantes dispersas.

A continuación, comparamos la regla LDA clásica con las técnicas discriminantes lineales de alta dimensión FAIR, PAM y DSDA. Para ello, dividimos la muestra original en dos partes: una de entrenamiento, sobre la cual se ajusta el modelo, y otra de testeo o validación, cuyas observaciones servirán para la predicción de las etiquetas de clase. La comparación de los métodos la haremos mediante el error o probabilidad de clasificación incorrecta, calculada como la fracción de etiquetas de clase predichas por el modelo que difieren de la verdadera etiqueta. Para reducir la variabilidad, repetimos el proceso de simulación 2000 veces y promediamos las estimaciones del error obtenidas.

En lo concerniente a su implementación en R, las reglas LDA y PAM se programan de forma directa. El método DSDA, por su parte, está basado en un modelo de regresión lineal con penalización Lasso, por lo que haremos uso del comando glmnet en R para aproximar su solución. Por último, debido a que FAIR no cuenta con un paquete disponible en R, tomaremos como referencia los resultados para el error de [10], los cuales, a pesar de haber sido obtenidos de una manera diferente, sirven para hacernos una idea del comportamiento de esta regla. La tabla 3.2 recoge la comparación de los métodos, donde el error está expresado en (%).

Como vimos en el capítulo 2, FAIR y PAM estiman la matriz de varianzas-covarianzas Σ mediante una matriz diagonal, con el fin de resolver el problema de la invertibilidad en alta dimensión ($p >> n$). Es decir, estas reglas ignoran la estructura de correlación entre las variables. Sin embargo, el método DSDA, basado en un modelo de regresión lineal con penalización Lasso, sí tiene en cuenta esta correlación y, por ello, es el único que muestra un buen comportamiento en los cinco modelos considerados. Los métodos PAM y FAIR tienen igualmente un comportamiento

	LDA	FAIR	PAM	DSDA
Modelo 1	68	11.47	2	0
Modelo 2	68	15.67	4	0
Modelo 3	37	25.69	35.5	0
Modelo 4	27.5	14.27	28	0
Modelo 5	29.5	24.14	29.5	0

Cuadro 3.2: Resultados de la simulación para cada modelo considerado.

aceptable, ya que su tasa de error se sitúa en todos los casos por debajo del 50 %. No obstante, el error es significativamente mayor que el de DSDA, a excepción del modelo 1.

En el modelo 1, los primeros cinco elementos de $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$ son mucho más grandes que el resto, lo que implica que las reglas basadas en la independencia de las variables pueden incluir las tres variables discriminantes (i.e, las variables del conjunto $\left\{j : \left\{\Sigma^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})_j \neq 0\right\}\right\}$). Por otro lado, aunque el modelo 2 usa la misma matriz de varianzas-covarianzas Σ que el modelo 1, tiene una estructura de medias muy diferente: los primeros dos elementos de $\boldsymbol{\mu_2} - \boldsymbol{\mu_1}$ son muy grandes, mientras que el resto son mucho más pequeños. Esto significa que las reglas de independencia tienen dificultades en seleccionar la variable 3, resultando en una peor clasificación.

Finalmente, los resultados de la simulación también muestran que LDA sigue siendo un clasificador competente aún en el contexto de la alta dimensión, pues su comportamiento se asemeja al de otras técnicas discriminantes como FAIR o PAM.

La Figura 3.1 muestra, gráficamente, el comportamiento de los cuatro métodos en los cinco modelos de simulación.

3.2. Datos reales

En el ámbito médico, una temprana detección de tumores malignos es difícil pero muchas veces crucial para un tratamiento exitoso. La información biomolecular es, a día de hoy, igual o incluso más importante para el diagnóstico del cáncer que los factores clínicos tradicionales. El reto reside en que algunos test, como los experimentos microarray, generan un gran conjunto de datos con valores para la expresión génica de cientos o miles de genes, pero con una muestra de muy pocos pacientes. Es decir, estamos en el caso de la alta dimensión, donde el número de variables es mucho mayor que la cantidad de observaciones disponibles.

Por tanto, para una comparación más exhaustiva de las diferentes reglas discriminantes,

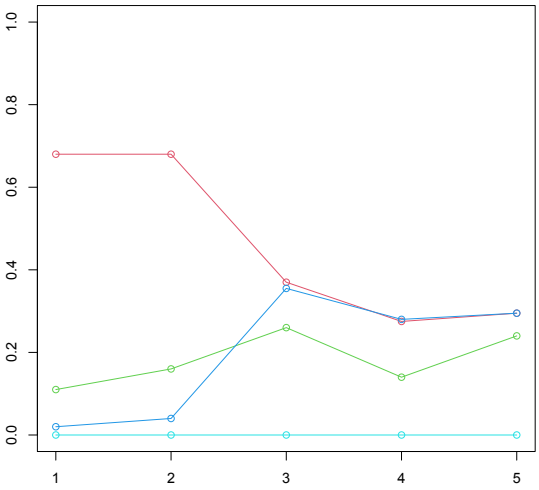

Figura 3.1: Error de clasificación medio para las distintas reglas sobre cada modelo de simulación (rojo: LDA, verde: FAIR, azul oscuro PAM, azul claro: DSDA)

haremos uso de dos bases de datos clásicas y de referencia en el contexto de la genética: la de cáncer de colon ([1]) y la de cáncer de próstata ([13]). El objetivo básico en estos casos es predecir si una nueva observación es o no tumoral.

La base de datos [1], disponible en R a través del paquete *HiDimDA*, está formada por 2000 genes (i,e 2000 variables) medidos sobre 62 pacientes (i.e, 62 observaciones): 40 de ellos diagnosticados con cáncer de colon y los 22 restantes sanos. El status del paciente está descrito por la variable factor *grouping* y los valores de los genes están dados por las variables numéricas "*genes.1*" hasta "*genes.2000*". Por su parte, el conjunto de observaciones [13], que está disponible en R a través del paquete *sda*, contiene mediciones de la expresión génica correspondiente a 6033 genes para una muestra de 102 observaciones: 52 son pacientes diagnosticados con cáncer de próstata y 50 son varones sanos.

En ambos casos, dividimos la muestra en una parte de entrenamiento y otra de validación con un ratio 2:1 (equivalentemente, 2/3 de la muestra será de entrenamiento y 1/3 de testeo). Repetimos este proceso 100 veces. Los porcentajes de error relativos a los distintos métodos en las dos bases de datos consideradas se recogen en la tabla 3.3, donde las probabilidades son las opuestas de las que aparecen en el artículo [10]. No implementamos esto en R puesto que su programación es análoga a la realizada para los datos simulados.

De forma global, observamos que los tres métodos tienen un comportamiento muy similar en la base de datos de colon, mientras que en la de próstata vemos, de nuevo, que DSDA es la mejor

	FAIR	PAM	DSDA
Colon	13.6	13.6	13.6
Próstata	23.5	8.8	5.9

Cuadro 3.3: Comparación de las reglas discriminantes con las bases de datos de colon y próstata.

regla discriminante. FAIR, por el contrario, es la que posee un mayor error de clasificación.

Finalmente, veremos cómo se comporta el clasificador Máquinas de Vector Soporte (MVS) en estas bases de datos reales y lo compararemos con la calidad de las reglas previamente mencionadas. Para ello, emplearemos el procedimiento en [3].

Como se describe en el artículo, previo al estudio aplicamos una transformación logarítmica en base 10 a las expresiones génicas para lograr un efecto de simetrización en las variables. Además, con el fin de prevenir que una sola observación domine el análisis, estandarizamos los datos. A continuación, de manera aleatoria, seleccionamos 200 genes (para reducir el número de variables) y particionamos la muestra, de forma que 2/3 de las observaciones sean de entrenamiento y 1/3 de validación. Los errores los calculamos, de nuevo, como la fracción de clases predichas por el modelo que difieren de la verdadera etiqueta.

Para construir el clasificador MVS, usamos la opción del kernel radial que ofrece R por defecto, pues es la que proporciona, generalmente, mejores resultados y solo necesita de dos parámetros: el parámetro de coste o penalización C, y el parámetro γ que aparece en la expresión del kernel radial. A efectos de mejor clasificación, ambos parámetros son escogidos por validación cruzada sobre la malla de valores $\{2^{-5}, 2^{-4}, ..., 2^{10}\} \times \{2^{-10}, 2^{-9}, ..., 2^{5}\}$ para cada conjunto de entrenamiento. Repitiendo este proceso 50 veces se obtienen los resultados de la tabla 3.4. En el apéndice se proporciona un esquema del código que se podría usar para llegar a tales valores en el conjunto de datos relativo al cáncer de colon.

	Colon	Próstata
MVS	15.05	7.88

Cuadro 3.4: Errores de clasificación MVS (%) en las bases de datos de colon y próstata, extraídos de [3]

Observamos que MVS arroja peores resultados que las reglas FAIR y PAM en la base de datos de colon, pero sin embargo en la de próstata tiene una tasa de error muy similar a la de DSDA que, como vimos antes, era la mejor regla discriminante. La Figura 3.2 proporciona la

comparación de los cuatro métodos en este último conjunto de datos.

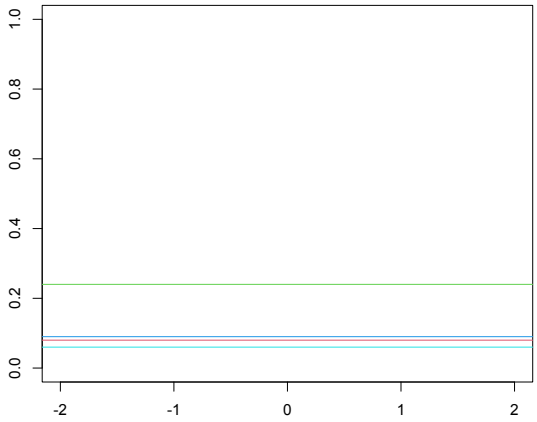

Figura 3.2: Error de clasificación promedio para las distintas reglas sobre la base de datos relativa al cáncer de próstata (rojo: MVS, verde: FAIR, azul oscuro PAM, azul claro: DSDA)

Por tanto, después de analizar cómo actúan los métodos de clasificación introducidos para el contexto del Big Data sobre varios conjuntos simulados y algunos reales, concluimos que aunque las reglas discriminantes basadas en la independencia de las observaciones suponen un gran atractivo a nivel computacional, pueden arrojar resultados no deseables en el aspecto de clasificación. Sin embargo, aquellos métodos como el DSDA, que sí tienen en cuenta la estructura de correlación entre las variables, poseen un comportamiento más homogéneo y son las que proporcionan las menores tasas de error. Por último, aunque el LDA clásico funciona peor que las técnicas previamente mencionadas, debido al efecto de la colinealidad entre variables, vemos que sigue teniendo un elevado poder de predicción.

Anexo I

Scripts utilizados para la implementación de los métodos

En este apéndice se recoge el código de R usado para generar los modelos de simulación, así como para implementar las reglas discriminantes LDA, PAM y DSDA. También se añaden las líneas de las Figuras 3.1 y 3.2, así como un esquema de código relativo a la obtención de los errores de clasificación por las Máquinas de Vector Soporte.

```
set.seed(2000)

library(MASS)
library(glmnet)
library(pamr)

#PARÁMETROS

#Número de observaciones
n1 = 100
n2 = 400

#Dimensiones
p1 = 400
p2 = 800

#Matrices de varianzas-covarianzas
```

```
Sig1 <- matrix(0,p1,p1)

for(i in 1:p1){
for(j in 1:p1){
Sig1[i,j] = 0.5^(abs(i-j))
}
}


Sig2 <- matrix(0,p2,p2)
for(i in 1:p2){
for(j in 1:p2){
if(i==j){
Sig2[i,j] = 1}
else{
Sig2[i,j] = 0.5
}
}
}

#GENERACIÓN MODELOS DE SIMULACIÓN

#MODELO 1

EstBayes1 = as.matrix(0.556*c(3,1.5,0,0,2,rep(0,p1-5))) #Estimador de Bayes
mu11 = as.matrix(rep(0,p1))    #Media para la primera clase
mu12 = Sig1%*%EstBayes1        #Media para la segunda clase
X1_clase1 = mvrnorm(n1/2,mu=mu11,Sigma=Sig1)
X1_clase2 = mvrnorm(n1/2,mu=mu12,Sigma=Sig1)

X1 = cbind(rbind(X1_clase1, X1_clase2), c(rep(1,n1/2), rep(2, n1/2))) #Modelo

#MODELO 2

EstBayes2 = as.matrix(0.582*c(3,2.5,-2.8,rep(0,p1-3)))
mu21 = as.matrix(rep(0,p1))
mu22 = Sig1%*%EstBayes2
```

```
X2_clase1 = mvrnorm(n1/2,mu=mu21,Sigma=Sig1)
X2_clase2 = mvrnorm(n1/2,mu=mu22,Sigma=Sig1)

X2 = cbind(rbind(X2_clase1, X2_clase2), c(rep(1,n1/2), rep(2, n1/2)))


#MODELO 3

EstBayes3 = as.matrix(0.395*c(3,1.7,-2.2,-2.1,2.55,rep(0,p2-5)))
mu31 = as.matrix(rep(0,p2))
mu32 = Sig2%*%EstBayes3

X3_clase1 = mvrnorm(n2/2,mu=mu31,Sigma=Sig2)
X3_clase2 = mvrnorm(n2/2,mu=mu32,Sigma=Sig2)

X3 = cbind(rbind(X3_clase1, X3_clase2), c(rep(1,n2/2), rep(2, n2/2)))

#MODELO 4

EstBayes4 = as.matrix(0.551*c(3,1.7,-2.2,-2.1,2.55,rep((1/(p2-5)),p2-5)))
mu41 = as.matrix(rep(0,p2))
mu42 = Sig2%*%EstBayes4

X4_clase1 = mvrnorm(n2/2,mu=mu41,Sigma=Sig2)
X4_clase2 = mvrnorm(n2/2,mu=mu42,Sigma=Sig2)

X4 = cbind(rbind(X4_clase1, X4_clase2), c(rep(1,n2/2), rep(2, n2/2)))

#MODELO 5

EstBayes5 = as.matrix(0.362*c(3,1.7,-2.2,-2.1,2.55,rep((1/(p2-5)),p2-5)))
mu51 = as.matrix(rep(0,p2))
mu52 = Sig2%*%EstBayes5

X5_clase1 = mvrnorm(n2/2,mu=mu51,Sigma=Sig2)
X5_clase2 = mvrnorm(n2/2,mu=mu52,Sigma=Sig2)
```

```
X5 = cbind(rbind(X5_clase1, X5_clase2), c(rep(1,n2/2), rep(2, n2/2)))

#-------------------------------

error.lda <- matrix(0,2000,5)
error.pam <- matrix(0,2000,5)
error.DSDA <- matrix(0,2000,5)

for(i in 1:2000){

#Tenemos que convertir los datos a data.frame para poder aplicar las reglas
discriminantes

N1 <- data.frame(X1)
N2 <- data.frame(X2)
N3 <- data.frame(X3)
N4 <- data.frame(X4)
N5 <- data.frame(X5)

#Dividimos la muestra en una parte de entrenamiento y otra de validación
#La mitad de observaciones serán de entrenamiento y la mitad de validación

train1 <- sample(1:n1, n1/2)
data.train1 <- N1[train1,]
data.test1  <- N1[-train1,]

data.train2 <- N2[train1,]
data.test2  <- N2[-train1,]

train2 <- sample(1:n2, n2/2)

data.train3 <- N3[train2,]
data.test3  <- N3[-train2,]

data.train4 <- N4[train2,]
data.test4  <- N4[-train2,]

data.train5 <- N5[train2,]
```

```
data.test5  <- N5[-train2,]

#CLASIFICACION LDA (Linear Discriminant Analysis)

Ajustamos modelo sobre la muestra de entrenamiento
clasificacion.lda1 <- lda(data.train1$X401~., data=data.train1)

#Predecimos la etiqueta de clase sobre la muestra de validación
pred.test.lda1 <- predict(clasificacion.lda1,data.test1)$class

#Calculamos la probabilidad de clasificación correcta (que mide la calidad de una
regla discriminante)
eficiencia.lda1<- mean(pred.test.lda1 == data.test1$X401)

#Calculamos la probabilidad de clasificacion incorrecta
error.lda1<- mean(pred.test.lda1 != data.test1$X401)


clasificacion.lda2 <- lda(data.train2$X401~., data=data.train2)
pred.test.lda2 <- predict(clasificacion.lda2,data.test2)$class
eficiencia.lda2<- mean(pred.test.lda2 == data.test2$X401)
error.lda2<- mean(pred.test.lda2 != data.test2$X401)

clasificacion.lda3 <- lda(data.train3$X801~., data=data.train3)
pred.test.lda3 <- predict(clasificacion.lda3,data.test3)$class
eficiencia.lda3<- mean(pred.test.lda3 == data.test3$X801)
error.lda3<- mean(pred.test.lda3 != data.test3$X801)

clasificacion.lda4 <- lda(data.train4$X801~., data=data.train4)
pred.test.lda4<- predict(clasificacion.lda4,data.test4)$class
eficiencia.lda4 <- mean(pred.test.lda4 == data.test4$X801)
error.lda4<- mean(pred.test.lda4 != data.test4$X801)

clasificacion.lda5 <- lda(data.train5$X801~., data=data.train5)
pred.test.lda5 <- predict(clasificacion.lda5,data.test5)$class
eficiencia.lda5 <- mean(pred.test.lda5 == data.test5$X801)
error.lda5<- mean(pred.test.lda5 != data.test5$X801)
```

```
#Almacenamos los errores de cada iteración
error.lda[i,] <- c(error.lda1,error.lda2,error.lda3,error.lda4,error.lda5)


#CLASIFICACION PAM (Nearest Shrunken Centroids Classifier)

clasificacion.pam1 <- pamr.train(list(x=as.matrix(t(data.train1)),...
y=as.factor(data.train1$X401)), threshold=1)
pred.test.pam1 <- pamr.predict(clasificacion.pam1, as.matrix(t(data.test1)),...
 threshold=1, type="class")
eficiencia.pam1<- mean(pred.test.pam1 == data.test1$X401)
error.pam1<- mean(pred.test.pam1 != data.test1$X401)

clasificacion.pam2 <- pamr.train(list(x=as.matrix(t(data.train2)),...
y=as.factor(data.train2$X401)), threshold=1)
pred.test.pam2 <- pamr.predict(clasificacion.pam2, as.matrix(t(data.test2)),...
 threshold=1, type="class")
eficiencia.pam2<- mean(pred.test.pam2 == data.test2$X401)
error.pam2 <- mean(pred.test.pam2 != data.test2$X401)

clasificacion.pam3 <- pamr.train(list(x=as.matrix(t(data.train3)),...
y=as.factor(data.train3$X801)), threshold=1)
pred.test.pam3 <- pamr.predict(clasificacion.pam3, as.matrix(t(data.test3)),...
 threshold=1, type="class")
eficiencia.pam3 <- mean(pred.test.pam3 == data.test3$X801)
error.pam3<- mean(pred.test.pam3 != data.test3$X801)

clasificacion.pam4 <- pamr.train(list(x=as.matrix(t(data.train4)),...
y=as.factor(data.train4$X801)), threshold=1)
pred.test.pam4 <- pamr.predict(clasificacion.pam4, as.matrix(t(data.test4)),...
 threshold=1, type="class")
eficiencia.pam4 <- mean(pred.test.pam4 == data.test4$X801)
error.pam4 <- mean(pred.test.pam4 != data.test4$X801)

clasificacion.pam5 <- pamr.train(list(x=as.matrix(t(data.train5)),...
y=as.factor(data.train5$X801)), threshold=1)
pred.test.pam5 <- pamr.predict(clasificacion.pam5, as.matrix(t(data.test5)),...
 threshold=1, type="class")
```

```
eficiencia.pam5 <- mean(pred.test.pam5 == data.test5$X801)
error.pam5 <- mean(pred.test.pam5 != data.test5$X801)

error.pam[i,] <- c(error.pam1,error.pam2,error.pam3,error.pam4,error.pam5)

#CLASIFICACION DSDA (Direct Sparse Discriminant Analysis)

res1 <- glmnet(as.matrix(data.train1),data.train1$X401,alpha=1)
#alpha=1 => PENALIZACION LASSO
#Escogemos el mejor parámetro de penalización lambda por validación cruzada usando
cv.glmnet(). Por defecto, este comando realiza validación validación cruzada de
10 iteraciones.
res.LA1 = cv.glmnet(as.matrix(data.train1),data.train1$X401,alpha=1,standardize=FALSE)
best.lambda1 <- res.LA1$lambda.min
#Predecimos con el mejor lambda
pred.test.DSDA1 <- round(predict(res1,s = best.lambda1, as.matrix(data.test1)),1)
eficiencia.DSDA1<- mean(pred.test.DSDA1 == data.test1$X401)
error.DSDA1<- mean(pred.test.DSDA1 != data.test1$X401)

res2 <- glmnet(as.matrix(data.train2),data.train2$X401,alpha=1)
res.LA2 = cv.glmnet(as.matrix(data.train2),data.train2$X401,alpha=1,standardize=FALSE)
best.lambda2 <- res.LA2$lambda.min
pred.test.DSDA2 <- round(predict(res2,s = best.lambda2, as.matrix(data.test2)),1)
eficiencia.DSDA2<- mean(pred.test.DSDA2 == data.test2$X401)
error.DSDA2<- mean(pred.test.DSDA2 != data.test2$X401)

res3 <- glmnet(as.matrix(data.train3),data.train3$X801,alpha=1)
res.LA3 = cv.glmnet(as.matrix(data.train3),data.train3$X801,alpha=1,standardize=FALSE)
best.lambda3 <- res.LA3$lambda.min
pred.test.DSDA3 <- round(predict(res3,s = best.lambda3, as.matrix(data.test3)),1)
eficiencia.DSDA3<- mean(pred.test.DSDA3 == data.test3$X801)
error.DSDA3<- mean(pred.test.DSDA3 != data.test3$X801)


res4 <- glmnet(as.matrix(data.train4),data.train4$X801,alpha=1)
res.LA4 = cv.glmnet(as.matrix(data.train4),data.train4$X801,alpha=1,standardize=FALSE)
best.lambda4 <- res.LA4$lambda.min
pred.test.DSDA4 <- round(predict(res4,s = best.lambda4, as.matrix(data.test4)),1)
```

```
eficiencia.DSDA4<- mean(pred.test.DSDA4 == data.test4$X801)
error.DSDA4<- mean(pred.test.DSDA4 != data.test4$X801)


res5 <- glmnet(as.matrix(data.train5),data.train5$X801,alpha=1)
res.LA5 = cv.glmnet(as.matrix(data.train5),data.train5$X801,alpha=1,standardize=FALSE)
best.lambda5 <- res.LA5$lambda.min
pred.test.DSDA5 <- round(predict(res5,s = best.lambda5, as.matrix(data.test5)),1)
eficiencia.DSDA5<- mean(pred.test.DSDA5 == data.test5$X801)
error.DSDA5 <- mean(pred.test.DSDA5 != data.test5$X801)

error.DSDA[i,] <- c(error.DSDA1,error.DSDA2,error.DSDA3,error.DSDA4,error.DSDA5)


}

#ERRORES DE CADA MÉTODO EN LOS 5 MODELOS CONSIDERADOS

colMeans(error.lda)
colMeans(error.pam)
colMeans(error.DSDA)


#GRÁFICO DE LOS ERRORES DE CLASIFICACIÓN (promediados)

error.fair <- c(0.11,0.16,0.26,0.14,0.24) #Error correspondiente a la regla FAIR

x <- 1:5
y <- seq(0,1,length.out=5)
plot(x,y,type="n",xlab='',ylab='')
lines(colMeans(error.lda),col=2,type='o') #Rojo: LDA
lines(error.fair,col=3,type='o') #Verde: FAIR
lines(colMeans(error.pam),col=4,type='o') #Azul oscuro: PAM
lines(colMeans(error.DSDA),col=5,type='o') #Azul claro: DSDA

#-------------------------------------

#MÁQUINAS DE VECTOR SOPORTE
```

```
#Bases de datos

library(HidimDA)
X <- AlonDS  #Base de datos cancer de colon

library(sda)
data(singh2002) #Base de datos cancer de próstata


#Hacemos solo el estudio para la base de datos de Alon (el otro caso se hace
igual, sustituyendo la base de datos)

levels(X$grouping) <- c("1", "2") #Cambiamos las etiquetas de las clases "colonc"
y "healthy" para que éstas sean números.

scale(log(X[,-1])) #Aplicamos una transformación logarítmica a los datos y los
estandarizamos, de forma que pertenezcan a una distribución normal de media 0
y varianza 1 (por tanto, las variables tomarán valores entre -2 y 2).

library(e1071) #Librería para la clasificación por Máquinas de Vector Soporte

error <- numeric(50)

for(i in 1:50){

#Preseleccionamos una muestra de 200 genes (al azar)

X.new <- X[,c(1,sample(1:2000,200))]

#Dividimos muestra: 2/3 muestra de entrenamiento y 1/3 muestra de validación
(aleatoriamente)

index     <- 1:nrow(X.new)
testindex <- sample(index, trunc(length(index)/3))
data.test   <- X.new[testindex,]
data.train  <- X.new[-testindex,]
```

```
#Máquinas de vector soporte

#Seleccionamos mejores parámetros por validación cruzada de un conjunto de posibles
valores sobre la muestra de entrenamiento

parametros <- tune.svm(data.train$grouping~., data = data.train,...
gamma = 2^(-10:5), cost = 2^(-5:10))

#Mejor modelo
svm.modelo <- parametros$best.model

#Predecimos sobre la muestra de validación, considerando el mejor modelo
svm.pred  <- predict(svm.modelo,data.train)

#Calculamos el error para cada iteración
error[i] <- mean(svm.pred != data.test$grouping)
}

error.definitivo <- mean(error) #Promediamos los valores de los errores obtenidos


#GRÁFICA ERRORES PROMEDIO EN LA BASE DE DATOS PRÓSTATA
x <- seq(-2,2,length.out=20)
y <- seq(0,1,length.out=20)
plot(x,y,type="n",xlab='',ylab='')
abline(h=0.08,col=2) #Rojo: MVS
abline(h=0.24,col=3) #Verde: FAIR
abline(h=0.09,col=4) #Azul oscuro: PAM
abline(h=0.06,col=5) #Azul claro: DSDA
```

Bibliografía

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*, Proceedings of the National Academy of Sciences - PNAS **96** (1999), no. 12, 6745–6750.

[2] C. Cortes and V. Vapnik, *Support-vector networks*, Machine Learning **20** (1995), 273–297.

[3] M. Dettling, *Bagboosting for tumor classification with gene expression data*, Bioinformatics **20** (2004), no. 18, 3583–3593.

[4] J. Fan and Y. Fan, *High dimensional classification using features annealed independence rules*, Annals of Statistics **36** (2008), no. 6, 2605–2637.

[5] R. A. Fisher, *The use of multiple measurements in taxonomic problems*, Annals of Eugenics **7** (1936), no. 2, 179–188.

[6] T. J. Hastie, R. J. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2nd ed., Springer, New York, 2009.

[7] G. Jamesa, D. Witten, T. J. Hastie, and R. J. Tibshirani, *An introduction to statistical learning: with applications in r*, 2nd ed., Springer, New York, 2021.

[8] I. Koch, *Analysis of multivariate and high-dimensional data*, Cambridge University Press, New York, 2014.

[9] Q. Mai, *A review of discriminant analysis in high dimensions*, WIREs Computational Statistics **5** (2013), no. 3, 190–197.

[10] Q. Mai, H. Zou, and M. Yuan, *A direct approach to sparse discriminant analysis in ultra-high dimensions*, Biometrika **99** (2012), no. 1, 29–42.

[11] Y. Qin, *A review of quadratic discriminant analysis for high-dimensional data*, WIREs Computational Statistics **10** (2018), no. 4, e1434.

[12] J. Shao, Y. Wang, X. Deng, and S. Wang, *Sparse linear discriminant analysis by thresholding for high dimensional data*, The Annals of Statistics **39** (2011), no. 2, 1241–1265.

[13] D. Singha, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers, *Gene expression correlates of clinical prostate cancer behavior*, Cancer Cell **1** (2002), no. 2, 203.

[14] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, *Diagnosis of multiple cancer types by shrunken centroids of gene expression*, Proceedings of the National Academy of Sciences - PNAS **99** (2002), no. 10, 6567–6572.