Universidad de **Valladolid**

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Estadística

PEC de grado en Estadística e Ingeniería Informática (INdat)

Métodos de selección de variables en clustering y análisis discriminante

Autor:

Luis Miguel Calvo Magaz

Universidad de **Valladolid**

Facultad de Ciencias

TRABAJO FIN DE GRADO

Grado en Estadística

PEC de grado en Estadística e Ingeniería Informática (INdat)

Métodos de selección de variables en clustering y análisis discriminante

Autor:

Luis Miguel Calvo Magaz

Tutor:

Agustín Mayo Íscar

A mi familia, gracias por darme la fuerza para conseguir mis objetivos

A mis padres, Luis e Irene y a mi hermana Irene, por no dejar nunca de creer en mí.

A Irene Morales por sacarme siempre una sonrisa en cualquier momento

Agradecimientos

Agradecer la elaboración de este trabajo a todos los profesores de Estadística de la Universidad de Valladolid que me han aportado las bases del conocimiento necesario para poder realizarlo.

A el ”TEAM” por hacer que disfrute mi vida universitaria más allá de los libros y los apuntes.

Resumen

Hoy en día es muy común en cualquier disciplina el querer identificar las características de un conjunto de individuos que permita diferenciar y separar a los mismos en dos o más grupos para posteriormente poder clasificar nuevos casos de individuos como pertenecientes a un grupo u otro, desde la medicina para saber si un medicamento va a ser beneficioso para cierto paciente dadas sus características físicas e historial clínico, a la educación para saber si cierto alumno va a ser capaz de aprobar una asignatura determinada dado su historial académico. Tradicionalmente esta discriminación en grupos se ha hecho en base a la experiencia u otros criterios poco fiables y sin ninguna razón de peso, por ello se han desarrollado técnicas estadísticas de **Análisis discriminante** que permiten detectar qué variables son realmente relevantes para la discriminación de los grupos y en qué medida. Sin embargo, a la hora de la práctica, existen muchas situaciones en las cuales el número de variables de los individuos es incluso superior al número de individuos, en estas condiciones los métodos tradicionales de análisis discriminante no son capaces de realizar buenas predicciones de clase puesto que tienen demasiadas variables explicativas. En estas situaciones son donde se pueden aplicar los **métodos de selección de variables** que prescindirían de las variables inútiles que sólo generarían ruido y las variables redundantes cuya información ya está explicada por otras variables del modelo, reduciendo así el número de variables del modelo a sólo las imprescindibles y la complejidad del mismo para que los métodos de análisis discriminante puedan realizar la separación eficientemente. En este trabajo de Fin de Grado se explorarán las diversas técnicas de análisis discriminante y selección de variables y se pondrán a prueba tanto teórica como prácticamente hasta encontrar los métodos que mejor funcionen según diversos escenarios propuestos.

Abstract

Nowadays it's very common in any discipline to want to identify the characteristics of a group of individuals that allows to differentiate and separate them into two or more groups to later classify new cases of individuals as belonging to one group or another, from medicine to know if a physic will be beneficial for a certain patient given their physical characteristics and clinical history, to education to know if a certain student will be able to pass a particular subject given their academic history. Traditionally this discrimination in groups has been done based on experience or other unreliable criteria and without any compelling reason, so statistical techniques have been developed **Discriminant analysis** that allow us to detect which variables are really relevant to the discrimination of groups and to what extent. However, when it comes to the practice, there are many situations in which the number of variables of individuals is even higher than the number of individuals, in these conditions the traditional methods of discriminant analysis are not capable of making good class predictions since they have too many explanatory variables. In these situations are where the **variable selection methods** can be applied that would dispense with useless variables that would only generate noise and redundant variables whose information is already explained by other variables of the model, thus reducing the number of variables of the model to only the essential ones and its complexity so that the methods of discriminant analysis can carry out the separation efficiently. In this Final Degree Project, the different techniques of discriminant analysis and variable selection will be explored and tested both theoretically and practically until the methods that work best according to different proposed scenarios are found.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Motivación

Hoy en día es muy frecuente para cualquier ámbito el tener que identificar las características de un conjunto de individuos que permita diferenciar y separar a los mismos en dos o más grupos para posteriormente poder clasificar nuevos casos de individuos como pertenecientes a un grupo u otro: ¿Conseguirá el alumno superar la asignatura X ? ¿Beneficiará el tratamiento Y al paciente? ¿devolverá este cliente el crédito prestado?, etc.

Si no se tienen las respuestas a estas preguntas, cualquier persona trataría de responderlas mediante su propia intuición o experiencia o la experiencia de otros, sin embargo, si la complejidad de las preguntas/problemas asciende, las consecuencias de una mala resolución pueden ser críticas y no valen respuestas o soluciones tomadas por intuición, sino soluciones constituidas por razones de peso y consistentes.

En este punto es donde entra el Análisis discriminante [1], una técnica estadística para la clasificación de los individuos en grupos identificando las variables que discriminarían a los individuos en los grupos de interés con la mayor precisión posible.

Sin embargo, esta técnica por si misma no siempre es eficiente, puesto que no siempre todas las variables explicativas para la función discriminante son realmente relevantes para diferenciar los grupos, o incluso que haya variables que estén siendo redundantes y estén aportando la misma información que otras que también están en el modelo, si se llega al caso extremo de que el número de variables supera al número de individuos, la mayoría de técnicas de análisis discriminante flaquean ya que no serían capaces de hacer ninguna predicción.

Para solucionar este problema se pueden realizar técnicas de selección de variables [2] para los procedimientos discriminantes de tal manera que reduciendo la dimensionalidad y el ruido que pueden aportar las variables innecesarias los métodos de análisis discriminante puedan hacer unas mejores discriminaciones y predicciones de grupos para nuevos individuos

En este trabajo se mostrarán distintos métodos de selección de variables aplicados a técnicas de análisis discriminante y se expondrán sus puntos fuertes y flaquezas desde el plano teórico hasta su implementación práctica en el lenguaje de programación R [3].

1.2. Objetivos

Este proyecto persigue cinco objetivos principales:

- Exposición de las técnicas de análisis discriminante más comunes y utilizadas.
- Descripción de los métodos Sparse para la selección de variables ya que se adaptan a la dispersión y son invariantes a la dimensionalidad.
- Presentación de métodos novedosos de selección de variables en clustering como el Random Forest.
- Implementación de los algoritmos de selección de variables en la práctica partiendo de la idea teórica.
- Descubrir qué métodos de selección de variables se desenvuelven mejor en la práctica para escenarios distintos.

1.3. Selección de variables

1.3.1. Introducción al análisis discriminante

El análisis discriminante [4] es una técnica estadística cuyo interés es el de clasificar a varios individuos o poblaciones en distintos grupos a partir de las diversas variables que esos individuos tienen. Cada uno de estos individuos sólo puede pertenecer a un grupo.

La variable que describe el grupo al que pertenece cada individuo se le denomina **variable respuesta o dependiente**, esta sería una **variable categórica** que toma tantos valores como grupos de individuos se discrimina.

En cuanto a las variables que ayudarían a discriminar estos grupos, se las denomina **variables explicativas o predictoras**. Con la ayuda de estas variables explicativas, las denominadas **funciones discriminantes** se encargarán de realizar las predicciones de las categorías para nuevos individuos que se desean clasificar.

En definitiva, El análisis discriminante busca **explicar la pertenencia** de cada individuo de la población en cada uno de los grupos predefinidos en función de las variables explicativas. [5] Y, a su vez, **cuantificar el peso de cada una de ellas en la discriminación**. Por otra parte, el análisis discriminante también persigue **predecir** cuál es grupo al que tiene mayor probabilidad de pertenecer una nueva observación conociendo únicamente sus variables explicativas.

1.3.1.1. Métodos tradicionales de análisis discriminante

1.3.1.1.1 Función discriminante de Fisher para la clasificación en dos grupos

Fisher propuso 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 los predictores que mejor explican la separación de las clases. Una vez encontradas dichas combinaciones se realiza la clasificación en este subespacio. Fisher definió como subespacio óptimo a aquel que maximiza la distancia entre grupos en términos de varianza. [6]

Se obtiene como función lineal de las k variables explicativas X:

$$D = u_1X_1 + u_2X_2 + ... + u_kX_k \tag{1.1}$$

Su objetivo es el de obtener los coeficientes u_j considerando las n observaciones, por tanto para un **individuo concreto i**, el valor de fisher para la observación i sería:

$$D_i = u_1X_{1i} + u_2X_{2i} + ... + u_kX_{ki} \quad i \in \{1..n\} \tag{1.2}$$

En notación matricial: $D = Xu$

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1k} \\ X_{21} & X_{22} & \dots & X_{2k} \\ \vdots & \vdots & \dots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{nk} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} \tag{1.3}$$

Calculando la variablidad de esta función mediante la suma de cuadrados de las variables con respecto a la media:

$$\begin{bmatrix} D_1 - \bar{d_1} \\ D_2 - \bar{d_2} \\ \vdots \\ D_n - \bar{d_n} \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1k} \\ X_{21} & X_{22} & \dots & X_{2k} \\ \vdots & \vdots & \dots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{nk} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} \tag{1.4}$$

$$D'D = u'X'Xu \tag{1.5}$$

Por tanto $T = X'X$ sería una matriz simétrica que expresa las desviaciones cuadráticas con respecto a la media de las variables, lo que sería la suma de cuadrados total.

Entonces esta matriz T simétrica de suma de cuadrados total se podría descomponer esta suma de cuadrados en dos:

- Suma de cuadrados DENTRO de los grupos (V)

- Suma de cuadrados FUERA de los grupos (F)

$$T = X'X = F + VD'D = u'X'Xu = u'(F + V)u = u'Fu + u'Vu \tag{1.6}$$

De esta ecuación podemos deducir que los ejes discriminantes entre ambos grupos vienen determinados por los autovectores asociados a los autovalores de la matriz $V^{-1}F$ ordenados de más discriminante a meno.

Entonces los valores D_i corresponderían con los valores obtenidos tras proyectar cada el individuo i para cada una de sus k variables sobre los ejes discriminantes.

Los coeficientes u se obtendrían maximizando la distancia entre grupos entre la distancia dentro de cada grupo, esto sería buscar una mayor separación FUERA de los grupos y una menor separación entre observaciones DENTRO de un mismo grupo, esto sería:

$$MAX\lambda = \frac{u'Fu}{w'Vu} \tag{1.7}$$

Centroides de los grupos (I, II):

$$\bar{X_I} = \begin{bmatrix} \bar{X_{1I}} \\ \bar{X_{2I}} \\ \vdots \\ \bar{X_{kI}} \end{bmatrix} \bar{X_{II}} = \begin{bmatrix} \bar{X_{1II}} \\ \bar{X_{2II}} \\ \vdots \\ \bar{X_{kII}} \end{bmatrix} \tag{1.8}$$

Para clasificar mediante esta función de Fisher bastaría con aplicar el siguiente criterio de clasificación:

$D_i < C$: Observación i pertenece al grupo I

$D_i > C$: Observación i pertenece al grupo II

Siendo C el punto de corte discriminante, en el caso de dos grupos:

$$C = \frac{\bar{D_I} + \bar{D_{II}}}{2} \tag{1.9}$$

1.3.1.1.2 Análisis discriminante lineal

El Análisis Discriminante Lineal o *LDA* de sus siglas en inglés (*Linear Discrimiant Analysis*) [7] es un método de clasificación supervisado de variables **cualitativas** en el que dos o más grupos son conocidos a priori y nuevas observaciones se clasifican en uno de ellos en función de sus características.

El LDA es una generalización del discriminante lineal de Fisher, Los términos de discriminante lineal de Fisher y LDA son a menudo usados para expresar la misma idea, sin embargo, el artículo original de Fisher describe un discriminante que no hace algunas de las suposiciones del LDA como son:

- Cada predictor que forma parte la función discriminante se **distribuye de forma normal** en cada una de las clases de la variable respuesta. En el caso de múltiples predictores, las observaciones siguen una distribución normal multivariante en todas las clases.
- La **varianza del predictor es igual en todas las clases de la variable respuesta**. En el caso de múltiples predictores, la matriz de covarianza es igual en todas las clases. En el caso de que esto no se cumpla se podrá recurrir al Análisis Discriminante Cuadrático (QDA).

Cabe destacar que, aunque la condición de normalidad no se cumpla y por lo tanto el LDA pierda precisión, puede llegar a clasificaciones relativamente buenas.

Existen varios enfoques posibles para realizar un LDA, aún que la aproximación de Fisher es la original y una de las más usadas, la aproximación que introduciré en este capítulo estará basada en el **teorema de Bayes**, donde se 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, mediante $P(Y = k|X = x)$. Se le asigna a la observación al grupo para la que tenga una mayor probabilidad predicha.

Teorema de Bayes para la clasificación LDA Teorema de Bayes [8] dados dos eventos A y B:

$$P(B|A) = \frac{P(AB)}{P(A)}$$

Suponiendo que se desea clasificar una nueva observación en uno de los K grupos de la variable cualitativa Y para K>=2 y con un único predictor. Probabilidad a priori de pertenencia al grupo k: π_k Función de densidad para una observación que pertenece a k: $f_k(X) \equiv P(X = x|Y = k)$. Probabilidad a posteriori de pertenencia al grupo k: $P(Y = k|X = x)$

Si se aplica el teorema de Bayes se tiene que:

$$P(Pertenecer\ a\ grupo\ k \mid valor\ x\ observado) = \frac{P(grupo\ k\ \&\ observar\ x)}{P(observar\ x)} = \tag{1.10}$$

$$P(Y = k|X = x) = \frac{\pi_k P(X = x|Y = k)}{\sum_{j=1}^{K} \pi_j P(X = x|Y = j} = \frac{\pi_k f_k(x)}{\sum_{j=1}^{K} \pi_j f_j(x)} \tag{1.11}$$

Para calcular la clasificación para esa observación que tenga menor error se tiene que buscar el grupo que MAXIMICE la probabilidad a posteriori: al ser equivalente el denominador $\sum_{j=1}^{K} \pi_j f_j(x)$ para todos los grupos, bastaría con maximizar en numerador: $MAX\pi_k f_k(x)$.

Clasificador de Bayes:

$$\hat{\delta_k}(x) = log(P(Y = k|X = x)) = x\frac{\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} + log(\pi_k) \tag{1.12}$$

Cabe destacar que, para que lo que se acaba de mencionar sea posible, se necesita conocer la **probabilidad poblacional de pertenencia a un grupo k** π_k y la **probabilidad poblacional de que dado que una observación pertenece al grupo k, tenga el valor de x en el predictor:** $(f_k(X)$, como en la práctica normalmente no se tienen estos datos, se suelen estimar a partir de la muestra, por ello el clasificador LDA que se obtiene **No es exactamente igual al clasificador de Bayes, se aproxima a él**:

$$\hat{\pi_k} = \frac{n_k}{N} \tag{1.13}$$

$$f_k(X) = P(Y = k|X = x) = \frac{1}{\sqrt{2\pi\sigma_k^2}} exp(\frac{-1}{2\sigma_k^2}(x-\mu_k)^2) \tag{1.14}$$

$$\hat{\mu_k} = \frac{1}{n_k}\sum_{i:y_1} x_i \quad \hat{\sigma_k} = \frac{1}{N-K}\sum_{k=1}^{K}\sum_{i:y_1}(x_i - \hat{\mu_k})^2 \tag{1.15}$$

$$\hat{f}_k(X) = P(Y = k|X = x) = \frac{1}{\sqrt{2\pi\hat{\sigma_k^2}}} exp(\frac{-1}{2\hat{\sigma_k}^2}(x-\hat{\mu_k})^2) \tag{1.16}$$

En el caso particular de una variable cualitativa Y, con solo dos niveles, se puede expresar la regla

de clasificación como un ratio entre las dos probabilidades a posteriori.

$$Si\ \frac{P(Y=1|X=x)}{P(Y=2|X=x)} > 1 \quad Asignar\ a\ grupo\ 1.$$
$$Si\ \frac{P(Y=1|X=x)}{P(Y=2|X=x)} \leq 1 \quad Asignar\ a\ grupo\ 2.$$
$$Lmite\ decision:\ x = \frac{\mu_1 + \mu_2}{2}$$

Esto se puede extrapolar para varios predictores añadiendo nuevas hipótesis:

- X es un vector de p predictores: $X_1, X_2, ..., X_p$
- Suponemos la distribución de X una distribución **Normal multivariante**.

Siguiendo el mismo procedimiento del LDA para un único predictor pero aplicando la ecuación multivariante normal y **asumiendo MISMA matriz de varianzas covarianzas** $\sum$ para todas las K clases, el clasificador de Bayes sería:

$$\hat{\delta_k}(x) = log(P(Y=k|X=x)) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + log(\pi_k) \tag{1.17}$$

E igual que en caso de un único predictor, si no se conocen los parámetros poblacionales se debe recurrir a la estimación de las medias, las probabilidades a priori y el Sigma: $\mu_1, ..., \mu_k$, $\pi_1, ..., \pi_k$ y Σ

1.3.1.1.3 Análisis discriminante cuadrático

Como se ha comentado en el apartado anterior, una de las limitaciones del análisis discriminante lineal, es, entre otras, que la matriz de varianzas y covariazas Σ debe ser la misma para los K grupos.

Sin embargo, el clasificador cuadrático o de sus siglas en inglés QDA (Quadratic Discriminat Analysis)[9] considera que cada clase k tiene su propia matriz de varianzas-covarianzas Σ_k, y por lo tanto, su función discriminante sería cuadrática:

$$\hat{\delta_k}(x) = log(P(Y=k|X=x)) = -\frac{1}{2}log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + log(\pi_k) \tag{1.18}$$

Al haber k matrices de varianzas-covarianzas se debe estimar para cada clase el Σ_k, μ_k y π_k a partir de la muestra como se ha hecho en el discriminante lineal.

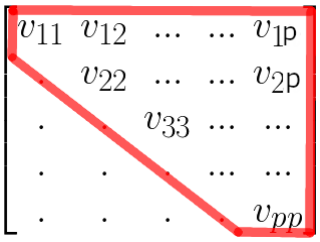
1.3.1.1.4 Comparación de discriminante lineal y cuadrático

Para elegir un clasificador u otro hay que tener en cuenta el balance de **bias-varianza** en el cual:

- **LDA** genera límites de decisión lineales, lo que produce una menor flexibilidad y un **menor problema de varianza**, pero si los grupos no son separables linealmente tendrá un **bias muy alto**.

- **QDA** genera límites de decisión cuadráticos, lo que aporta una mayor flexibilidad y ajuste a los dato, esto producirá un **bias menor**, sin embargo habrá un **mayor riesgo de varianza**.

LDA funciona mejor con pocas observaciones, ya que en este tipo de situaciones es muy importante evitar la varianza. Sin embargo, para un **número mayor de individuos**, no se puede asumir que varios grupos tengan una matriz de varianzas-covarianzas igual, por lo que sería mejor optar por **QDA**. También hay que tener en cuenta el coste computacional de estimación de los predictores, en el caso de que se elija un discriminante lineal, bastaría con estimar la ÚNICA matriz de varianzas covarianzas:



Número de predictores a estimar= $\frac{p\times(p+1)}{2}$

Sin embargo, para un discriminante cuadrático habría que estimar K matrices de varianzas-covarianzas lo que sería un total de $K \times \frac{p\times(p+1)}{2}$
Para valores de p muy altos, la elección de la función discriminante podría estar limitada por la capacidad de cómputo.

1.3.2. Introducción a la selección de variables en el análisis discriminante

En la práctica se suele disponer de un conjunto grande de variables explicativas para la función discriminante, sin embargo, ¿todas esas variables están aportando información relevante para esa función? ¿puede que alguna o la combinación de alguna esté aportando la mima información que otra? En caso de que la respuesta sea negativa, convendría suprimir alguna de esas variables que no aportan nada a la función y que lo hacen más complejo puesto que, como se ha visto en la sección anterior, el número de predictores p, es clave en el coste computacional, ya que a mayor número de predictores, mayor será la carga computacional del método de clasificación.

La intuición nos diría que, a mayor número de variables predictoras, mayor será la capacidad predictiva, y por lo tanto, mejores serán las predicciones, sin embargo, a mayor número de variables, mayor es la cantidad de parámetros a estimar y por tanto su precisión individual será menor (mayor variabilidad).

Tampoco interesa una función discriminante con menos variables de las necesarias, ya que se podría obtener una función mal detallada y obtener estimaciones de los grupos sesgadas.

Los métodos de selección de variables [10] resolverían este problema ya que:

- Reducen el ruido que generan las variables irrelevantes.
- Evitarían el almacenamiento innecesario de esas variables.
- Mejorarían el rendimiento del algoritmo de clasificación, a menor número de predictores, menor número de parámetros a estimar.

1.3.2.1. Selección de variables

Todos los algoritmos de selección de variables tienen en cuenta dos aspectos importantes:

- **Redundancia**: Una variable es redundante si está correlacionada en gran medida con otras de las variables, lo que significa que está aportando la misma información que estas.
- **Relevancia**: Una variable se considera relevante si es realmente útil para descubrir los grupos de las observaciones.

Estos algoritmos de selección de variables suprimirán las variables irrelevantes y las redundantes.

Para medir la relevancia se suele hacer un análisis descriptivo univariante calculando las medias y las desviaciones típicas de las variables en cada uno de los grupos y, si para alguna de estas, las diferencias entre las medias de los grupos es grande y la variabilidad es pequeña, se la considera potencialmente relevante para discriminar.

Una vez se tengan esas variables, se observan las relaciones entre las mismas para comprobar si existe alguna redundancia, para ello, se calculan las matrices de correlaciones a partir de las de covarianzas para una mejor interpretabilidad, se analiza la correlación entre pares de variables sin distinguir grupos y posteriormente, analizar las correlaciones dentro de cada grupo y calcular la media de los mismos.

Para comparar las variables y medir cuales pueden ser más interesantes se usan los siguientes estadísticos:

- **F de Snedecor**: Compara para cada variable las desviaciones de las medias de cada grupo a con la media total del grupo, entre las desviaciones a la media DENTRO de cada grupo.
  - Si **F es grande** para las variables, las medias están muy separadas en cada grupo, esto quiere decir que la variable discrimina bien.
  - Si **F es pequeña**, hay heterogeneidad entre los grupos o están demasiado próximos por lo que no discriminará bien.
- λ **de Wilks o Estadísitco U**: Se utiliza para selección de variables individuales, sin tener en cuenta las demás, λ es igual al cociente entre la suma de cuadrados DENTRO de los grupos y la suma de cuadrados TOTAL (Desviaciones a la media DENTRO de cada grupo entre desviaciones a la media TOTAL): $\frac{SumaCuadradosDENTROgrupo}{SumaCuadradosTOTAL\ (Sin\ distinguir\ grupos)}$
  - Si λ **es grande** La variabilidad total se debe a las diferencias DENTRO de los grupos, lo que sería malo para la discriminación, malas clasificaciones
  - Si λ **es pequeña**, La variabilidad total se debe a las diferencias ENTRE grupos, lo que sería ideal para las clasificaciones.

1.3.2.2. Métodos de selección de variables

Fundamentalmente se suelen usar dos métodos de selección de variables:

El **método directo** que consiste en escoger todas las variables originales si cumplen alguno de los criterios de selección, por ejemplo, que el λ sea menor que un valor predefinido. Este método no es óptimo puesto que **no tiene en cuenta las relaciones** que pueden haber entre las variables y por lo tanto se podría caer en la **redundancia**.

El **método *stepwise*** en el cuál se van seleccionando las variables por iteraciones, los pasos que sigue el método serían los siguientes:

1.3.2.2.1 Método *stepwise* para la selección de variables

1. Se incluye la variable con el mejor valor aceptable para el criterio de selección o de **entrada**. (Por ejemplo la variable con menor λ).
2. Se evalúan las demás variables no seleccionadas según el criterio y se selecciona la que tenga el mejor valor.
3. Una vez se haya introducido esta variable se evalúa todas las variables seleccionadas y se comprueba según un **criterio de salida** si deben seguir en la función discriminante. En el caso de que cumplan el criterio de salida se extraerá de la función.
4. Repetir desde el paso 2 hasta que no queden variables que cumplan el criterio de entrada (2) o el criterio de salida(3). ->Función dicriminante óptima.

Al paso 4 se le considera "criterio de parada", sin embargo no siempre se llega a la función discriminante óptima en un tiempo razonable, por ello se suele implementar un **número máximo de iteraciones del método** ya que una misma variable puede ser seleccionada y eliminada en bucle infinitas veces.

Por lo general este número máximo de iteraciones se suele fijar al **doble del número de variables originales**.

Adicionalmente se añade un **mecanismo de "protección frente a la redundancia"** añadiendo al paso 2 una comprobación de la **tolerancia**:

Se define R_i como el coeficiente de correlación múltiple para las p variables originales, este expresa la variabilidad de cada variable $X_i, ..., X_p$ recogida por el resto de las $p-1$, siendo $R_i \in 0, 1$ donde $R_i = 0$ significa que la variabilidad o información que recoge la variable i no la recoge ninguna otra variable y donde $R_i = 1$ significa que la variabilidad que podría aportar la variable i en e la función discriminante ya está recogida por el resto de variables. lógicamente interesa Coeficientes de correlación múltiples lo más próximos a 0. Se define también R_i^2 como el coeficiente de determinación, y $1 - R_i^2$ **como la tolerancia**, de esta manera cuanto **mayor sea la tolerancia** de una variable, **mayor será la información** INDEPENDIENTE que aporta a la función.

Por lo tanto, si la tolerancia de una de las variables que se ha seleccionado para incluir a la función discriminante es demasiado baja con respecto al resto de variables ya incluidas en la función, esta variable no será incluida.

1.3.2.2.2 Criterios de entrada y de salida

Para la determinación de los criterios de entrada y de salida de las variables en la función discriminante, primero se debe calcular los estadísticos F y $\lambda\ de\ Wilks$ multivariantes:

$$F = \frac{|B|}{|W|} \tag{1.19}$$

donde:
$|B|$ es el determinante de la matriz de varianzas-covarianzas ENTRE grupos.
$|W|$ es el determinante de la suma de las matrices de varianzas-covarianzas DENTRO de los grupos.

A partir del valor de F se puede obseter el valor de la $\lambda\ de\ Wilks$:

$$F = \frac{n-k-p-1}{k-1}(\frac{1}{\lambda} - 1) \tag{1.20}$$

$$\lambda = ((\frac{k-1}{n-k-p-1}F) + 1)^{-1} \tag{1.21}$$

donde:
n es el número de observaciones.
k es el número de grupos.
p es el número de variables.

Estos dos estadísticos se interpretan igual que en caso univariante:
Si **F es grande** para las variables, las medias están muy separadas en cada grupo, esto quiere decir que la variable discrimina bien.
Si λ **es pequeña**, La variabilidad total se debe a las diferencias ENTRE grupos, lo que sería ideal para las clasificaciones.

Entonces, lo estadísticos que se utilizarían en el procedimiento *stepwise* serían:

- Estadístico **F de entrada**: Expresa la disminución de la $\lambda\ de\ Wilks$ cuando se incluye una nueva variable a la función discriminante, si el valor de la F es pequeño implicaría que la disminución de la $\lambda\ de\ Wilks$ es inaceptable y por tanto no se puede permitir que esa variable entre en la función.

- Estadístico **F de salida**: Expresa el incremento de la $\lambda\ de\ Wilks$ cuando se elimina una de las variables que ya están en la función discriminante, si el valor de la F es pequeño implicaría que el incremento de la $\lambda\ de\ Wilks$ no es suficientemente significativo por lo que esa variable no es relevante en la función.

Todo esto tiene sentido si los grupos son realmente diferentes unos de otros, esto es que hayan diferencias significativas entre ellos, sin embargo, si estos grupos son prácticamente iguales no tendría sentido intentar separarlos de ninguna manera.

Para determinar si hay diferencias significativas entre los grupos se podría realizar un contraste de hipótesis estimando la $\lambda\ de\ Wilks$ total mediante el producto de las $\lambda\ de\ Wilks$ de cada función discriminante, esto seguiría una χ^2:

$$V = -(n - 1 - \frac{p + k}{2})ln(\lambda) \tag{1.22}$$

Donde $V \sim \chi^2_{p(k-1)}$,
V sigue una distribución χ^2 con $p(k - 1)$ grados de libertad, de modo que, si λ es pequeño, V será grande y por tanto la hipótesis se rechazara.

Para ser más concreto, si el valor de V calculado mediante $V = -(n - 1 - \frac{p+k}{2})ln(\lambda)$ es menor que $\chi^2_{0.05,p(k-1)}$ (el valor de la $\chi^2_{p(k-1)}$ a nivel α = 0.05), se rechazaría la hipótesis nula con una confianza del 95 %

Capítulo 2

Métodos Sparse de selección de variables

En la práctica, existen muchas situaciones en las cuales el número de variable es superior o bastante superior que el número de observaciones $p > n$ o $p >> n$. Esto es un gran problema para los métodos de clasificación tradicionales como el de *Fisher*, *LDA* o *QDA* vistos en la introducción puesto que la matriz de varianzas-covarianzas es singular.

Cuando $p >> n$ necesitamos un clasificador que seleccione variables. Este permite una interpretación más fácil del modelo y reducir el ajuste excesivo de los datos de entrenamiento, pero no vale un método convencional de selección de variables como sería el *stepwise* puesto que reduciría la dimensionalidad del problema y por lo tanto la matriz de varianzas-covarianzas resultante no sería singular.

Ahí es donde entrarían los llamados **métodos Sparse** [11] los cuales se adaptan a la dispersión y usan métodos de penalización para reducir la dimensionalidad de las variables, estas penalizaciones pueden ser de varios tipos: Ridge l_2 (No sparsity), Lasso l_1 y de red elástica, que combina las penalizaciones l_1 y l_2 de los método de Lasso y Ridge.

Penalización l_2 de Ridge

Cuando $p >> n$, el $rango(X^{-T}X) \leq n << p$, por lo tanto, $X^{-T}X$ es necesariamente singular y el estimador de mínimos cuadrados fallaría, por ello, en la regresión de Ridge [12] se toma:

$$\hat{\beta}_{n,\lambda} = argmin_{\beta \in \mathbb{R}^p} ||Y - XB||_2^2 + \lambda ||\beta||_2^2 \tag{2.1}$$

Y el método de penalización:

$$\hat{\beta}_{n,\lambda} = ((X^T X)^{-1} X^T + \lambda I)^{-1} X^T Y \tag{2.2}$$

Descomposición en valores singulares de $X^T X$:

$$X = UDV^T \tag{2.3}$$

Donde U es ortogonal, al igual que V y $D = diag(d_1, .., d_p)$ con $d_1 \geq d_2 \geq ... \geq d_p$, entonces:

$$X^T X = UD^2V^T \; donde \; D^2 = diag(d_1^2, .., d_p^2) \tag{2.4}$$

$$\hat{Y}_\lambda = X\hat{\beta}_{n,\lambda} = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}(u_j^T y)u_j \tag{2.5}$$

Entonces, la regresión Ridge "encoge" las coordenadas respecto de sus componentes principales normalizadas, a **mayor "encogimiento" para las componentes principales, se tiene una menor varianza**.

Cambiando los regresores X por sus estimaciones $\tilde{X} = UD(Y = \tilde{X}\tilde{\beta}_0 + \epsilon, \tilde{\beta}_0 = V^T\beta_0)$

$$E(\hat{\beta}_{n,j,\lambda} - \tilde{\beta}_j) = (sesgo(\hat{\beta}_{n,j,\lambda}))^2 + VAR(\hat{\beta}_{n,j,\lambda}) \tag{2.6}$$

donde:

$$\hat{\beta}_{n,j,\lambda} = \frac{d_j}{d_j^2 + \lambda}u_j^T Y \; ; \; VAR(\hat{\beta}_{n,j,\lambda}) = \frac{d_j^2}{(d_j^2 + \lambda)^2}\sigma^2 \tag{2.7}$$

$$E(\hat{\beta}_{n,j,\lambda} - \tilde{\beta}_j) = \tilde{\beta}_j{}^2 \frac{\lambda^2}{(d_j^2 + \lambda)^2} + \frac{d_j^2}{(d_j^2 + \lambda)^2}\sigma^2 \tag{2.8}$$

Por lo tanto, la regresión Ridge reduciría su Error Cuadrático Medio si:

$$\frac{d_j^2(d_j^2 + \lambda^2 \frac{\tilde{\beta}_j}{\sigma^2})}{(d_j^2 + \lambda)^2} \leq 1 \tag{2.9}$$

Entonces, λ , d_j y $\frac{\tilde{\beta}_j^2}{\sigma^2}$ sería los paráemtros que reducirían y aumentarían el valor del ECM.

El problema de este método Ridge es que no selecciona automáticamente las variables, a diferencia de la **Regresión Lasso**:

Penalización l_1 de Lasso

El problema de Lasso [12] sería **equivalente al de Ridge** pero para un $t > 0$:

$$\hat{\beta}_{n,\lambda} = argmin_{\beta:||\beta||_1 \leq t}||Y - XB||_2^2 \tag{2.10}$$

Sin embargo, para que este método funcione bien hay que elegir adecuadamente el t (λ), si se elige mal este parámetro se puede tender al sobreajuste del problema. El problema es que λ de parámetros desconocidos. Por lo general se estima λ por validación cruzada partiendo la muestra en K grupos

elegidos al azar y fijado uno de ellos para test y el resto para train. Posteriormente se estima el E.C.M y se repite para los K grupos. El λ que consiga una menor estimación del E.C.M será el elegido.

A continuación, en la figura 2.1 se muestra el algoritmo de descenso de coordenadas aplicando la penalización de Lasso.

Método iterativo. En el paso j se actualiza β_j

1. Iterar hasta la **convergencia** o hasta un número **máximo de iteraciones** predefinido (Descenso por coordenadas cíclico):

(a) For $j = 1, ..., p$:

(i.) Minimizar β_j:

$$\min_{\beta_j} \sum_{i=1}^{n} (y_i - \sum_{k \neq j} x_{ik}\beta_k - x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.11}$$

(ii.) Calculo del residuo parcial $r_i^{(j)}$:

$$r_i^{(j)} = y_i - \sum_{k \neq j} x_{ik}\beta_k \tag{2.12}$$

(iii.) Minimizar β_j:

$$\min_{\beta_j} \sum_{i=1}^{n} (r_i^{(j)} - x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.13}$$

(iv.) Actualización de $\hat{\beta}_j$:

$$\hat{\beta}_j = S_\lambda(x_j \cdot r^{(j)}) \tag{2.14}$$

Figura 2.1: Algoritmo Lasso

Como $\hat{\beta}_j = 0$ para algunos j, se puede utilizar el método de Lasso para la **selección de variables** definiendo los índices:

$$\hat{J} = \{j : \hat{\beta}_j \neq 0\} \tag{2.15}$$

Entonces se puede probar que el índice de las variables más importantes:

- Con alta probabilidad (Sobre $X^T X$):

$$\hat{J} \subset J_0 = \{j : {\beta^*}_j \neq 0\} \tag{2.16}$$

- Con alta probabilidad (β^* verdadero vector de coeficientes):

$$\frac{1}{n}||X(\hat{\beta} - \beta_0)||_2^2 \leq \frac{C}{k^2}\frac{\sigma^2 s_0 log(p)}{n} \quad donde\ s_0 = card(J_0) \tag{2.17}$$

El problema de este método Lasso es que no maneja bien que las variables estén correladas a diferencia del Ridge, entonces, ¿Es posible mantener una selección de variables y que se pueda manejar

bien las variables correladas? La respuesta es SI, gracias a las redes elásticas:

Penalización de redes elásticas

Las redes elásticas [12] combinan las penalizaciones de Lasso y Ridge regulándolas mediante un parámetro α para el cual:

$$\alpha = 0 \quad Ridge. \tag{2.18}$$

$$\alpha = 1 \quad Laso. \tag{2.19}$$

La función objetivo cambia:

$$\min_{\beta_0,\beta} \frac{1}{2}\sum_{i=1}^{n} ||y_i - (\beta_0 + x_i'\beta)||^2 + \lambda[\frac{1}{2}(1-\alpha)||\beta||_2^2 + \alpha||\beta||_1] \tag{2.20}$$

Si $\alpha \in (0,1)$ intervalo cerrado, se mantiene el criterio 'sparsity' y funciona bien (o al menos mejor), con variables correladas.

$$\hat{\beta}_j = \frac{S_{\lambda\alpha}(\sum_{i=1} nr_{ij}x_{ij})}{\sum_{i=1} nx_{ij}^2 + \lambda(1-\alpha)} \tag{2.21}$$

Residuos: $r_{ij} = y_i - \hat{\beta}_0 - \sum_{k\neq j} x_{ij}\hat{\beta}_k$

El criterio de elección del α es el mismo que el del λ del Lasso, por validación cruzada se calcularía el óptimo.

Uno de los métodos más conocidos de selección de variables en regresión es el denominado **GLMnet**[13] en el cuál se realiza un modelo de regresión múltiple (GLM) aplicándole una penalización por red elástica (Elastic net con penalizaciones l_1 y l_2

2.1. Sparse Linear Discriminant Analysis (Sparse LDA)

En los capítulos introductorios ya se ha hablado del Discriminante Lineal (LDA) en el cual:

Suponiendo **X** una matriz $n \times p$, suponer que cada una de las n observaciones pertenece a una de los **K** grupos. también suponer que las **p** variables están estandarizadas y centradas en 0 (media 0 e igual varianza).
Siendo x_i la observación **i** y C_K los índices de la observación en la clase K.
Considerar un modelo Multinormal donde una observación de la clase K sigue una distribución $N(\mu_k, \Sigma_k)$ donde $\mu_k \in R^p$ es la media del grupo K y Σ_k es la matriz de Varianzas-Covariazas dentro del grupo K.
Una estimación para μ_k sería $\frac{1}{|C_k|}\sum_{i\in C_k} x_i$.
Una estimación para Σ_k sería $\frac{1}{n}\sum_{k=1}^{K}\sum_{i\in C_k}(x_i-\mu_k)(x_i-\mu_k)^T$.

El LDA estimaría la clase más similar a la observación de Test aplicando la ley de Bayes.

Sin embargo este procedimiento como se ha explicado antes no funciona bien cuando:

- El número de variables predictoras es mayor que el número de observaciones, esto sería que la matriz de varianzas-covarianzas dentro del grupo es SINGULAR.
- Una sola distribución normal por clase no es suficiente para realizar la separación.
- Los datos no son separables linealmente.

El SparseLDA [14] supondría que:
Siendo **X** una matriz $n \times p$,e **y** un vector de longitud **n**, se puede resolver el problema de la dimensionalidad utilizando métodos penalizados.

Lasso resuelve el problema:

$$minimize_{\beta}\{|y - X\hat{\beta}|^2 + \lambda|\beta|_1\} \tag{2.22}$$

Otra opción es usar una penalización de Red elástica l_2:

$$minimize_{\beta}\{|y - X\hat{\beta}|^2 + \lambda|\beta|_1 \ + \nu|\beta|_1^2\} \tag{2.23}$$

Siendo λ y ν parámetros de tunning.
Para λ grandes tanto Lasso como la red elástica producirán estimaciones de vectores de coeficientes dispersos.
Sin embargo La penalización l_2 parece mejor que la l_1 porque las variables correlacionadas tienden a asignarse coeficientes de regresión similares, y se pueden incluir más de $min(n, p)$ variables en el modelo. Se usará una penalización l_2.
Ya existen muchas propuestas de extensión al LDA, sin embargo ninguna de ellas trabaja con clasificadores DISPERSOS.

El algoritmo SDA que producirá la regla de clasificación de los datos se muestra en la figura 2.2:

2.1.1. Ejemplos

Para probar la potencia de este método frente a otros se propone una simulación de observaciones normales con los siguientes datos:

En primer lugar se realizará una discriminación para **dos grupos**, para ello se generarán dos muestras de normales bivariantes con la misma matriz de varianzas-covarianzas para cada uno de los dos grupos, para este ejemplo se generarán:
100 observaciones de Normales $N(\begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})$ para el **grupo 1**.

1. Dada la matriz de respuesta Y $n \times K$. Inicialmente: $Y_{ij} = 1_{i \in C_k}$

2. Dado $D_\pi = \frac{1}{n} Y^T Y$.

3. Inicialmente $k = 1$ y la matriz Q_1 $K \times 1$, matriz de 1s.

4. $for\ k = 1, ..., q$ calcular un nuevo par de direcciones SDA (θ_k, β_k) de la siguiente manera:

(a) Inicializar $\theta_k = (I - Q_k Q_k^T D_\pi)\theta_*$ donde θ_* es un vector K-dimensional aleatorio:

(i.) Normalizar θ_k de tal manera que $\theta_k = \theta_k^T D_\pi \theta_k = 1$

(b) Iterar hasta la **convergencia** o hasta un número **máximo de iteraciones** predefinido:

(i.) β_k sería la solución al problema generalizado de la red elástica:

$$MINIMIZE_{\beta_k}\{\frac{1}{n}||Y\theta_k - X\beta_k||^2 + \gamma\beta_k^T\Omega\beta_k + \lambda||\beta_k||_1\} \tag{2.24}$$

(ii.) Para el β_k ajustado, tomar:

$$\hat{\theta}_k = (I - Q_k Q_k^T D_\pi) D_\pi^{-1} Y^T X \beta_k\ ,\quad \theta_k = \hat{\theta}_k / \sqrt{\hat{\theta}_k^T D_\pi \hat{\theta}_k} \tag{2.25}$$

(c) Si $k < q$, entonces $Q_{k+1} = (Q_k : \theta_k)$

5. El algoritmo LDA para la matriz $n \times q$ $(X\beta_1\ X\beta_2\ ...\ X\beta_q)$ daría como resultado la regla de clasificación.

Figura 2.2: Algoritmo Sparse LDA

100 observaciones de Normales $N(\begin{pmatrix}2\\2\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix})$ para el **grupo 2**.

En la figura 2.3 se puede observar la distribución de estos grupos en las dos variables generadas.

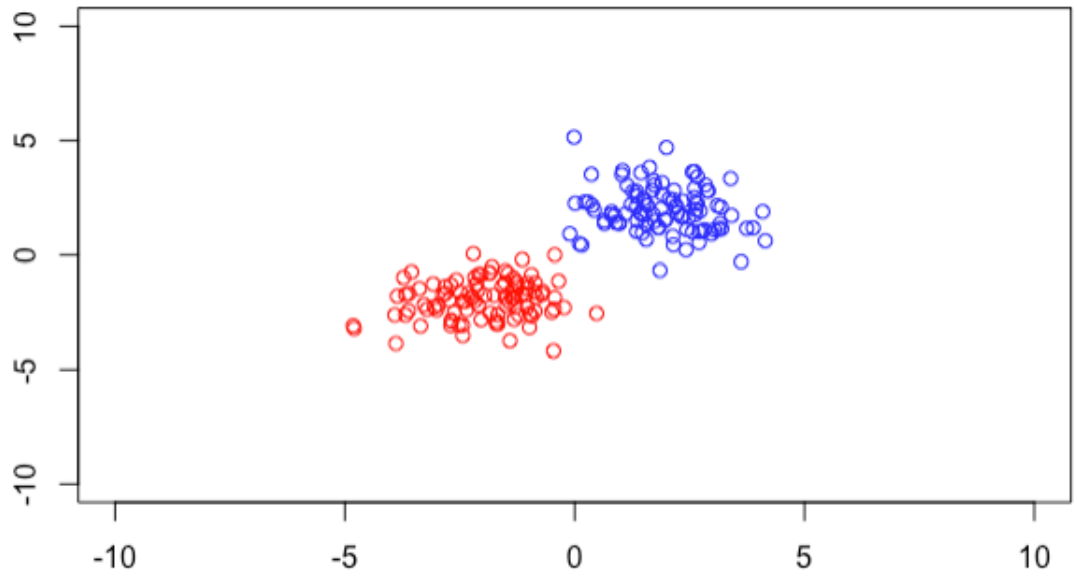


Figura 2.3: Distribución de los grupos en las dos variables.

Se puede observar perfectamente como esos dos grupos son separables linealmente 2.4 y un LDA con esos dos grupos únicamente podría realizar una discriminación perfecta si se seleccionan aleatoriamente $2/3$ de las observaciones (134 observaciones) para entrenamiento y el $1/3$ restante para test (66 observaciones)

```
          Reference
Prediction Grupo 1 Grupo 2
   Grupo 1      37       0
   Grupo 2       0      29
Accuracy
       1
```

Figura 2.4: Matriz de confusión en el conjunto test para la función LDA definida

El LDA proyectaría las observaciones tests con dos variables en observaciones con una única variable **LD1** 2.5 que separa perfectamente ambos grupos:

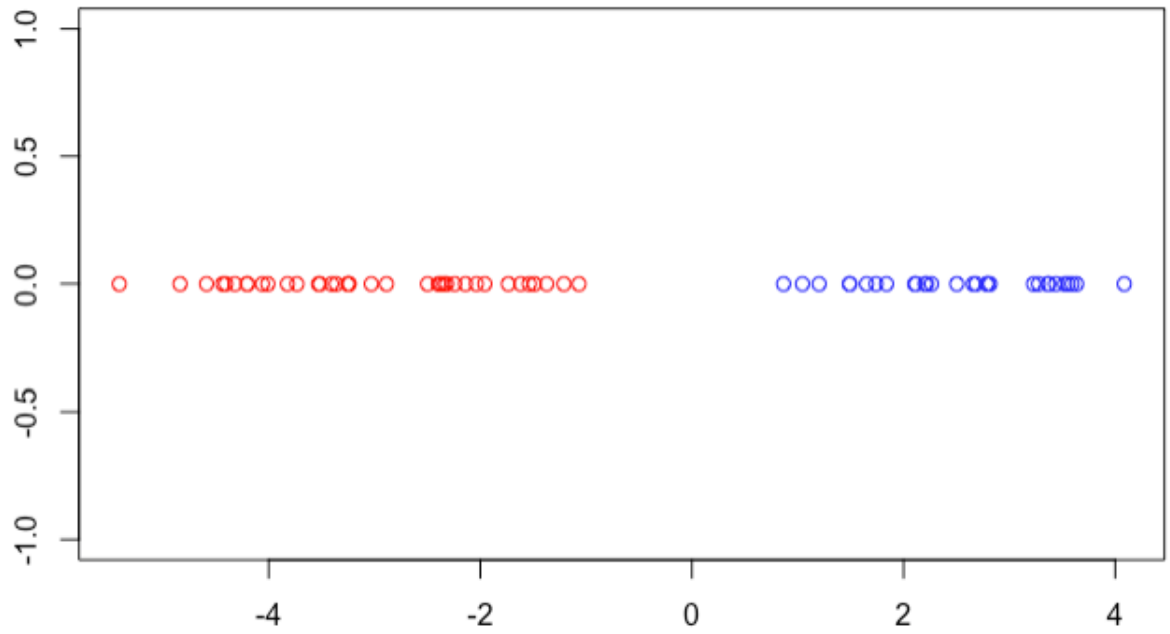


Figura 2.5: Proyecciones de las observaciones en la variable LD1

Sin embargo, si se generan 100 variables aleatorias para generar ruido en los datos, por ejemplo, 100 variables nuevas de $N(0,1)$, los resultados serían los que se muestran en las figuras 2.6 y (2.7, la función no es capaz de discriminar bien los datos para tantas dimensiones, como es lógico, por ello se debe recurrir a un procedimiento de selección de variables que encuentre las variables que podrían separar los dos grupos.

	Reference	
Prediction	Grupo 1	Grupo 2
Grupo 1	22	14
Grupo 2	15	15

Accuracy
0.5606061

Figura 2.6: Matriz de confusión para la función LDA con variables de ruido

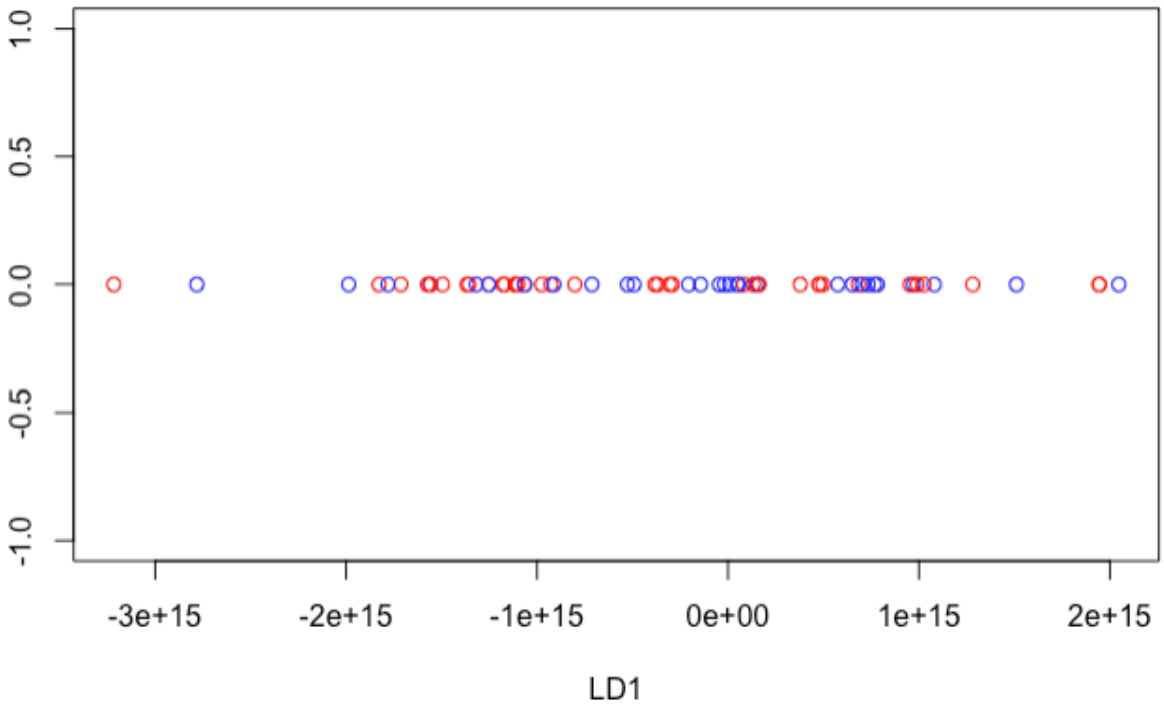


Figura 2.7: Proyecciones de las observaciones en la variable LD1 con variables de ruido

Aplicando el método **Sparse LDA** definido anteriormente las predicciones vuelven a ser perfectas, puesto que este método encuentra las variables que aportan una mayor información para discriminar ambos grupos, en este caso las dos variables generadas originalmente, la 101 y la 102 y realiza un LDA con esas dos variables como se ha hizo antes de introducir el ruido.

Vemos por lo tanto que la utilidad de este método Sparse, más que la de predecir los grupos, es la de seleccionar las variables que aportan realmente la información, ya que las predicciones las hace mediante un LDA tras la selección.

2.2. Sparse LDA con mezclas de normales

Sin embargo, este método tiene ciertas limitaciones como son:

1. Cuando las medias de los grupos están cerca.
2. Cuando la matriz de varianzas-covarianzas es muy amplia.
3. Cuando los K grupos no son separables por límites de decisión lineales.
4. Cuando las matrices de ambos grupos son perpendiculares.

Las dos primeras son limitaciones de cualquier método discriminante ya que, cuánto más parecidos son los dos grupos de datos, menos se podrán diferenciar.
El tercero es una limitación de cualquier método discriminante basado en hiperplanos que separan linealmente los grupos, ya que, si un solo prototipo por clase es insuficiente para capturar la estructura de la clase, entoces el rendimiento del LDA sería bastante pobre.

Y por último, la cuarta limitación es por definición de este método de discriminación ya que se basa en que las matrices de varianzas-covarianzas sean la misma y por ende, la peor situación que podría darse sería que ambas matrices fuesen totalmente perpendiculares.

Cuando las variables que contienen la información que podría separar los dos grupos son totalmente perpendiculares 2.8 el método Sparse LDA no es capaz de diferenciar los grupos puesto que las matrices de varianzas-covarianzas de las variables que separan esos grupos son lo más diferentes posibles, y lo más importante, no es capaz de seleccionar esas variables 2.9 .

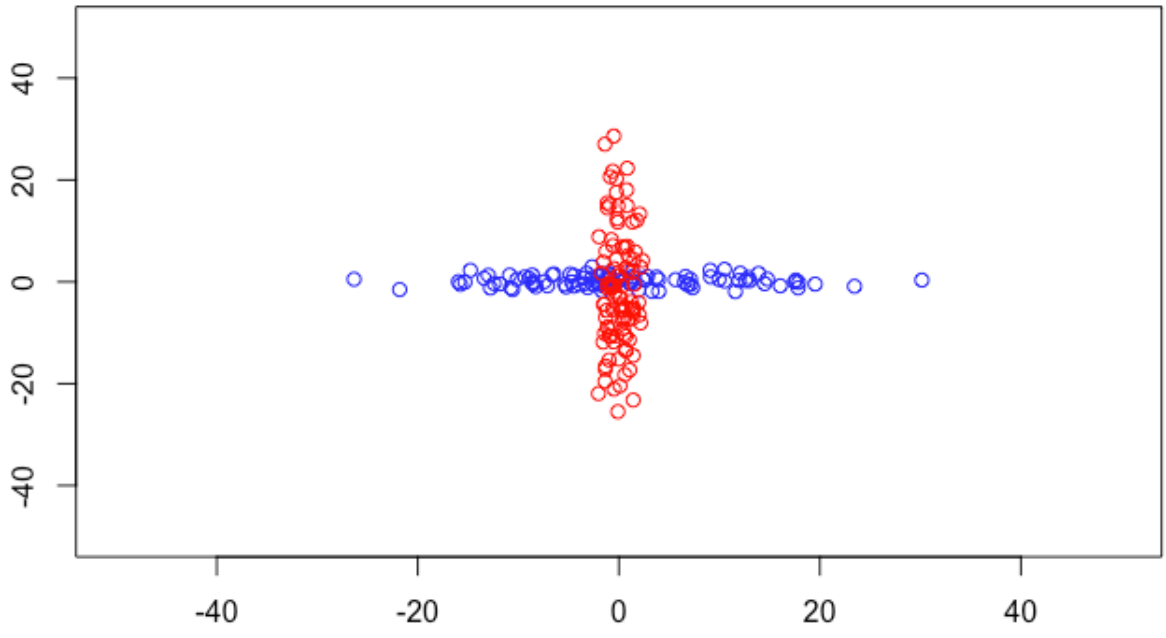


Figura 2.8: Variables 101 y 102 que contienen la información que podría separar los grupos

```
Prediction Grupo 1 Grupo 2
   Grupo 1      19      27
   Grupo 2       7      13
 Accuracy
0.4848485
[1] "Mejores variables: "
[1] 109 196
```

Figura 2.9: Matriz de confusión para la función LDA con las variables seleccionadas

Sin embargo, este método Sparse podría tener potencial para encontrar esas variables si en vez de modelar las observaciones dentro de cada clase como normales multivariantes con un vector medio específico de cada clase y una matriz de varianzas-covarianzas común dentro de las clases, se toma un modelo en el cual cada clase se distribuye como una mezcla de normales y así alcanzar una mayor flexibilidad para el modelo, de tal manera que, la clase o grupo k para $k = 1, ..., K$ se divide en subclases R_k. Esto es lo que propone el modelo **Sparse LDA con mixturas de normales**.

2.2.1. Introducción al análisis discriminante por Mixturas

Como ya se ha mencionado anteriormente, una de las debilidades del LDA, como de cualquier método discriminante basado en hiperplanos consiste en que si los K grupos no pueden ser separados linealmente (un sólo prototipo por clase es insuficiente para captar la estructura de la clase), entonces el el LDA tendrá un rendimiento pobre. Para solucionar esto se propone realizar un **análisis discriminante por mezcla de normales** (MDA) [15] para superar las deficiencias que supone el LDA en este punto.

Definiendo $R = \sum_{k=1}^{K} R_k$, se supone que el subgrupo r dentro del grupo k, $r = 1, ..., R_k$ tiene una distribución normal multivariante con una media específica para cada subclase $\mu_{kr} \in \mathbb{R}^p$ y una matriz de varianzas-covarianzas $p \times p$ común dentro del subgrupo Σ_w.

Suponiendo Π_k como la probabilidad a priori de pertenencia al grupo k y π_{kr} como la probabilidad de "mezcla" para la subclase r de la clase k con $\sum_{r=1}^{R_k} = 1$. Se puede estimar fácilmente Π_k si se tienen los datos, sin embargo lo que no es trivial es la estimación de las probabilidades de "mezcla" π_{kr}.

Para estimar estas probabilidades de "mezcla" o probabilidades a priori de los subgrupos, se podría utilizar el Algoritmo EM(Hastie y Tibshirani - 1996 [16]) empleando la .Expectativa-Maximización" para estimar los vectores medios específicos para cada subclase, la matriz de varianzas-covarianzas "WHITIIN"class (Dentro del grupo) y las probabilidades de "mezcla" para cada subgrupo:

2.2.2. Algoritmo EM para la estimación de la probabilidades de mezcla en cada subgrupo

Paso 1: Estimación de la probabilidad de que la observación i pertenezca al subgrupo r de la clase k dado que ya pertenece a la clase k:

$$p(c_{kr}|x_i, i \in C_k) = \frac{\pi_{kr} exp(-(x_i - \mu_{kr})^T \Sigma_w^{-1}(x_i - \mu_{kr})/2}{\sum_{r'=1}^{R_k} \pi_{kr'} exp(-(x_i - \mu_{kr'})^T \Sigma_w^{-1}(x_i - \mu_{kr'})/2)},\ r = 1, ..., R_k \tag{2.26}$$

En 2.26 se formula el evento de que la observación x_i se encuentra en la subclase r de la clase k.

Paso 2: En el **paso de maximización**, se actualizan la estimaciones para la probabilidades de "mezcla" de cada subclase 2.27 así como los vectores medios específicos de cada subclase 2.28 y las matrices de varianzas-covarianzas DENTRO de la clase 2.29 se calcularían:

$$\pi_{kr} = \frac{\sum_{i \in C_k} p(c_{kr}|x_i, i \in C_k)}{\sum_{r'=1}^{R_k} \sum_{i \in C_k} p(c_{kr'}|x_i, i \in C_k)} \tag{2.27}$$

$$\mu_{kr} = \frac{\sum_{i \in C_k} x_i p(c_{kr}|x_i, i \in C_k)}{\sum_{i \in C_k} p(c_{kr}|x_i, i \in C_k)} \tag{2.28}$$

$$\Sigma_w = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{r=1}^{R_k} p(c_{kr}|x_i, i \in C_k)(x_i - \mu_{kr})(x_i - \mu_{kr})^T \tag{2.29}$$

El algoritmo EM iteracionará entre las ecuaciones 2.26 y 2.29 hasta que se llegue a la convergencia.

2.2.3. Análisis discriminante de mezcla dispersa (Sparse Mixture Discriminant Analysis) SMDA

Se define una matriz Z de respuesta "borrosa" $n \times R$ que sería la matriz de probabilidades de pertenencia a la subclase, en el caso de que la observación i pertenezca a la la clase k, la fila i de la matriz Z contendrá las probabilidades de valores: $p(c_{k1}|x_i, i \in C_k), ..., p(c_k R_k|x_i, i \in C_k)$ en la columna k de las entradas R_k y 0 en las demás posiciones.

Z sería la matriz de mezcla análoga a la matriz de respuesta Y pero con las subclases.

Para realizar este análisis discriminante de mezcla dispersa se extiende el algoritmo MDA de la sección 2.2.1 realizando un SDA (Sparse Discriminant Analysis) descrito en la sección 2.1, tomando la matriz Z en lugar de Y como matriz de respuesta.

Y, en lugar de realizar las actualizaciones del algoritmo EM 2.26...2.29 sobre los datos X sin procesar, se utilizarán los datos transformados XB donde:

$$B = (\beta_1 ... \beta_q) \tag{2.30}$$

y donde $q < R$, esta matriz XB servirá como una proyección q-dimensional de los datos.

El algoritmo SMDA [17] que producirá la regla de clasificación de los datos se muestra en la figura 2.10:

1. Inicializar las probabilidades de cada subclase $p(c_{kr}|x_i, i \in C_k$ para cada instancia realizando R_k-medias clusteing sin la clase k.

2. Usar las probabilidades de las subclases calculadas en el paso 1 para crear la matriz de respuestas $n \times R$ Z

3. Iterar hasta la **convergencia** o hasta un número **máximo de iteraciones** predefinido:

(a) Usar Z en lugar de Y y realizar un SDA para encontrar la secuencia de $q < R$ pares de (vectores de puntuación , vectores discriminantes): $\{\theta_k, \beta_k\}_{k=1}^{q}$

(b) Calcular $\hat{X} = XB$ donde $B = (\beta_1 ... \beta_q)$

(c) Calcular los pesos de las medias, matriz de varianzas-covarianzas, y las probabilidades de "mezcla" usando las ecuaciones del método EM 2.27 ... 2.29 sustituyendo $\hat{X}$ por X:

$$\pi_{kr} = \frac{\sum_{i \in C_k} p(c_{kr}|\hat{x}_i, i \in C_k)}{\sum_{r'=1}^{R_k} \sum_{i \in C_k} p(c_{kr'}|\hat{x}_i, i \in C_k)} \tag{2.31}$$

$$\hat{\mu}_{kr} = \frac{\sum_{i \in C_k} \hat{x}_i p(c_{kr}|\hat{x}_i, i \in C_k)}{\sum_{i \in C_k} p(c_{kr}|\hat{x}_i, i \in C_k)} \tag{2.32}$$

$$\hat{\Sigma}_w = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{r=1}^{R_k} p(c_{kr}|\hat{x}_i, i \in C_k)(\hat{x}_i - \mu_{kr})(\hat{x}_i - \mu_{kr})^T \tag{2.33}$$

(d) Calcular las probabilidades de cada subgrupo utilizando la ecuación del método EM 2.26 sustituyendo $\hat{X}$ por X y usando las estimaciones de los pesos de las medias, matriz de varianzas-covarianzas y probabilidades de "mezcla" calculadas en el paso (c):

$$p(c_{kr}|x_i, i \in C_k) = \frac{\pi_{kr} exp(-(\hat{x}_i - \hat{\mu}_{kr})^T \hat{\Sigma}_w^{-1} (\hat{x}_i - \hat{\mu}_{kr})/2}{\sum_{r'=1}^{R_k} \pi_{kr'} exp(-(\hat{x}_i - \hat{\mu}_{kr'})^T \hat{\Sigma}_w^{-1} (\hat{x}_i - \hat{\mu}_{kr'})/2)}, \; r = 1, ..., R_k \tag{2.34}$$

(e) Usar las probabilidades a posteriori de las subclases calculadas en el paso (d) para actualizar la matriz respuesta Z

4. La regla de clasificación del algoritmo SMDA consistirá en asignar la nueva observación $x_{test} \in \mathbb{R}^p$ a el grupo para el que

$$\Pi_k \sum_{r=1}^{R_k} \pi_{kr} exp(-(\hat{x}_i - \hat{\mu}_{kr})^T \hat{\Sigma}_w^{-1} (\hat{x}_i - \hat{\mu}_{kr})/2 \tag{2.35}$$

sea el mayor.

Figura 2.10: Algoritmo SMDA

Contra todo pronóstico, este método no es capaz de encontrar las dos variables que contienen la información aún que se haga un procedimiento con mixtura de normales indiferentemente del número

de grupos como se puede observar en las figuras 2.11 2.12 2.13 2.14 2.15

Las probabilidades a priori de pertenencia a los subgrupos son prácticamente equivalentes, al igual que las probabilidades a posteriori para las observaciones del conjunto test, por lo tanto, el método está discriminando aleatoriamente debido a que no ha sido capaz de encontrar las variables que contienen la información para separar ambos grupos.

```
          Reference
Prediction Grupo 1 Grupo 2
   Grupo 1      30      27
   Grupo 2       3       6
 Accuracy
0.5454545
[1] "Mejores variables: "
[1] 103 124
```

Figura 2.11: Matriz de confusión para el SMDA y las variables seleccionadas

```
Prior probabilities of groups:
Grupo 1.1 Grupo 1.2 Grupo 2.1 Grupo 2.2
0.2164179 0.2835821 0.2910448 0.2089552
```

Figura 2.12: Probabilidades a priori estimadas para 4 subgrupos

```
$subprob
      Grupo 1.1 Grupo 1.2 Grupo 2.1 Grupo 2.2
 [1,] 0.2497258 0.2502745 0.2494095 0.2505902
 [2,] 0.2487011 0.2512994 0.2473509 0.2526487
 [3,] 0.2435392 0.2564619 0.2369897 0.2630092
 [4,] 0.2477454 0.2522551 0.2454313 0.2545682
 [5,] 0.2547949 0.2452047 0.2595897 0.2404107
 [6,] 0.2477381 0.2522624 0.2454166 0.2545828
 [7,] 0.2531353 0.2468645 0.2562581 0.2437420
 [8,] 0.2510856 0.2489146 0.2521412 0.2478587
 [9,] 0.2478516 0.2521489 0.2456446 0.2543549
[10,] 0.2471524 0.2528482 0.2442404 0.2557590
[11,] 0.2502706 0.2497296 0.2505039 0.2494958
[12,] 0.2493401 0.2506602 0.2486347 0.2513650
[13,] 0.2482992 0.2517013 0.2465436 0.2534560
[14,] 0.2472368 0.2527638 0.2444099 0.2555896
```

Figura 2.13: Probabilidades a posteriori estimadas para un subconjunto de las observaciones test en 4 subgrupos

```
Prior probabilities of groups:
 Grupo 1.1  Grupo 1.2  Grupo 1.3  Grupo 1.4  Grupo 1.5  Grupo 2.1  Grupo 2.2  Grupo 2.3  Grupo 2.4  Grupo 2.5
0.11940299 0.06716418 0.06716418 0.12686567 0.11940299 0.08208955 0.16417910 0.10447761 0.09701493 0.05223881
```

Figura 2.14: Probabilidades a priori estimadas para 10 subgrupos

```
      Grupo 1.1  Grupo 1.2  Grupo 1.3  Grupo 1.4  Grupo 1.5  Grupo 2.1  Grupo 2.2  Grupo 2.3  Grupo 2.4  Grupo 2.5
 [1,] 0.10041106 0.09999465 0.09880498 0.10037719 0.10041137 0.09992263 0.10009430 0.10004708 0.10012734 0.09980940
 [2,] 0.10066244 0.10074040 0.09780988 0.10032864 0.10045803 0.09970851 0.10046039 0.09998606 0.10035101 0.09949464
 [3,] 0.10188806 0.10453041 0.09290255 0.10003524 0.10064387 0.09862705 0.10231433 0.09966919 0.10147488 0.09791442
 [4,] 0.10089436 0.10143750 0.09688858 0.10028046 0.10049860 0.09950876 0.10080221 0.09992863 0.10055933 0.09920157
 [5,] 0.09912926 0.09634054 0.10382703 0.10056919 0.10013267 0.10097843 0.09829370 0.10033958 0.09901880 0.10137081
 [6,] 0.10089615 0.10144293 0.09688145 0.10028008 0.10049890 0.09950721 0.10080486 0.09992818 0.10056095 0.09919930
 [7,] 0.09955601 0.09753079 0.10216413 0.10051520 0.10023274 0.10063328 0.09888148 0.10024550 0.09938221 0.10085865
 [8,] 0.10007357 0.09900907 0.10013542 0.10043657 0.10034447 0.10020630 0.09960977 0.10012704 0.09983042 0.10022738
 [9,] 0.10086871 0.10135997 0.09699063 0.10028595 0.10049423 0.09953096 0.10076421 0.09993503 0.10053619 0.09923411
[10,] 0.10103719 0.10187132 0.09631956 0.10024915 0.10052237 0.09938465 0.10101473 0.09989269 0.10068860 0.09901975
[11,] 0.10027643 0.09959939 0.09933644 0.10040167 0.10038526 0.10003629 0.09990008 0.10007924 0.10000844 0.09997675
[12,] 0.10050598 0.10027506 0.09842965 0.10035930 0.10042931 0.09984206 0.10023200 0.10002419 0.10021154 0.09969090
[13,] 0.10076020 0.10103316 0.09742192 0.10030873 0.10047542 0.09962458 0.10060399 0.09996199 0.10043859 0.09937143
[14,] 0.10101688 0.10180941 0.09640056 0.10025368 0.10051904 0.09940235 0.10098441 0.09989783 0.10067017 0.09904567
```

Figura 2.15: Probabilidades a posteriori estimadas para un subconjunto de las observaciones test en 10 subgrupos

Capítulo 3

Método Random Forest para la selección de variables

3.1. Introducción a Random Forest

Un Random Forest [18] o ”Bosque aleatorio” en español es un método discriminante que consiste en la combinación o ensemble de árboles predictores de tal manera que cada árbol dependa de los valores de un único vector aleatorio independiente de cada árbol.

El algoritmo para inducir estos árboles combina la idea de un algoritmo de bagging [19] y la selección aleatoria de variables o atributos o Random subspace [20].

Para entender bien la idea del Random Forest es importante primero entender el Bagging:

El **Bagging** consiste esencialmente en promediar muchos modelos ruidosos pero imparciales para obtener un buen modelo que no tenga excesiva variación. 3.1:

1. En primer lugar se debe **dividir** el conjunto de Train en **varias muestras aleatorias** con el mismo número de individuos y con remplazamiento de estos.
2. Crear un **modelo predictivo** para cada muestra aleatoria, en este caso, árboles de decisión
3. Promediar todos los modelos anteriormente creados y ensamblarlos en uno único.

Figura 3.1: Algoritmo Bagging

El Random Forest es una variante del bagging que utiliza árboles de decisión aleatorios como modelos predictivos, lo que es perfecto para este tipo de algoritmo, puesto que los árboles pueden registrar estructuras de interacción complejas en los datos, y si crecen suficientemente profundo, tienen relativamente baja parcialidad, y dado que los árboles son bastante ruidosos, se benefician enormemente al promediar.

En el Random Forest, en cada nodo se muestrean aleatoriamete y sin remplazamiento M atributos de los K atributos totales, y de esos M atributos, **seleccionar el mejor**, por lo general el valor recomendado de M es de $log_2(K+1)$.

Random Forest introduce diversivad, puesto que combina:

- Muestreo con remplazamiento del conjunto de entrenamiento. (Bagging)
- Muestreo sin remplazamiento del conjunto de atributos. (Random Forest)

Algoritmo Random Forest (3.2):

1. Dado **N individuos** y **M número de variables** en el clasificador
2. Dado **m número de variables de entrada** para determinar la decisión de un nodo. ($m << M$).
3. Elegir un conjunto train para cada árbol y usar el resto de individuos como conjunto test.
4. Para cada nodo de cada árbol, elegir aleatoriamente m variables de las M originales para formar el modelo de decisión. Y calcular la mejor partición del conjunto train a partir de esas m variables.

Figura 3.2: Algoritmo Random Forest

Para la predicción de un nuevo individuo se empuja a través del árbol y se le etiqueta según el nodo terminal donde acabe. Este proceso se repite para TODOS los árboles del ensamblaje y se le asigna a la observación el grupo para el cual tenga una mayor cantidad de incidencias.

3.2. Random Forest aplicado a la selección de variables

Random Forest ya realiza implícitamente una selección de variables, sin embargo puede llegar a ser poco precisa e inestable, por ello en esta sección se hará una propuesta de un procedimiento de dos pasos en el cuál se promediarán las mejores variables que detecten varias ejecuciones de Random Forest Anidadas sobre un mismo conjunto de entrenamiento.

A continuación se detalla el algoritmo propuesto basado en Random Forest para la selección de variables mediante un procedimiento de dos pasos (two-steps procedure) (3.3):

Paso 1. Eliminación preliminar de variables y cálculo de un "ranking":

- Hacer un Ranking de las variables ordenándolas por VI (Medida de importancia de la variable "Variable Importance") Promediando varias ejecuciones de Random Forest en orden descendiente (Por ejemplo 50).
- Eliminar las variables que se consideren de poca importancia. (m denotaría el número de variables restantes).
  - Partiendo del ranking, considerar la secuencia ordenada de deviaciones típicas (sd) de VI y utilizar estas para estimar un valor umbral para el VI, l**as variables con un Vi mayor o igual al umbral serán seleccionadas**. También se puede establecer un número de variables elegidas como límite.
  - Dado que la variabilidad de VI es mayor para las variables reales que para las inútiles, el valor umbral viene dado por una estimación de la desviación típica de VI de las variables inútiles. Este umbral se ajusta al valor de predicción mínimo dado por un modelo CART(Modelo formado por árboles de decisión) en el que Y es la sd del VI, y X son las filas.

Paso 2. Selección de variables:

- **Para la interpretación**:
  - **Construir una colección anidada de modelos de Random Forest que involucre las primeras k variables** para $k = 1..m$ y seleccionar las variables involucradas en el modelo que generen el menor error OOB (Out-Of-Bag error [21] que estima el error de predicción del modelo Random Forest). Esto implicaría **considerar m' variables en lugar de m**.
    Siendo más preciso, se calcula las tasas de error OOB para todos los modelos RF empezando por el modelo que sólo tiene una única variable importante hasta el que incluya todas las variables importantes previamente seleccionadas.
    Idealmente se seleccionan las variable que conducen a un menor error OOB, sin embargo este método **puede ser inestable**, para combatir esta inestabilidad, se propone seleccionar el **modelo más pequeño con un error OOB menor que el error OOB mínimo aumentado por su desviación típica**.
- **Para la predicción**:
  - **Contruir una secuencia ascendente de modelos Random Forest** invocando y probando las variables seleccionadas en la parte de interpretación de forma escalonada.
    Siendo más preciso, Sólo se añade la variable si la disminución del error es mayor que el umbral definido. La idea es que la disminución del error OOB debe ser significativamente mayor que la variación media obtenida añadiendo variables de ruido.
    Ese umbral se fija mediante la media de los valores absolutos de los errores OOB de primer orden diferenciados entre el modelo con m' variables y el modelo con m variables:

$$\frac{1}{m - m'} \sum_{j=m'}^{m-1} |errorOOB(j+1) - errorOOB(j)| \tag{3.1}$$

    Donde $errorOOB(i)$ es el error OOB del modelo Random Forest en esa ejecución para las i variables más importates.

Figura 3.3: Algoritmo Random Forest para selección de variables

Capítulo 4

Comparativa de los métodos

En este capítulo se pondrán a prueba los métodos explicados en los capítulos anteriores y se procederá a realizar una comparativa de los mismos para algunas de las infinitas situaciones en las que se podrían aplicar, concretamente para este capítulo se valorará el desempeño de los métodos de selección de variables y análisis discriminante para datos simulados.

4.1. Situación 1: Dos grupos, dos variables con la información y la misma matriz de varianzas-covarianzas.

Se van a generar dos muestras de normales bivariantes con la misma matriz de varianzas-covarianzas para cada uno de los dos grupos, con esto se obtendrán las dos variables que serán capaces de discriminar ambos grupos:

100 observaciones de Normales $N(\begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix})$ para el **grupo 1**.

100 observaciones de Normales $N(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix})$ para el **grupo 2**.

En la figura 4.1 se puede observar la distribución de estos grupos en las dos variables generadas.

Se simulan posteriormente 200 variables aleatorias a mayores para generar ruido en los datos, en este caso, variables $N(0,1)$, un ejemplo de representación de estas variables se muestra en la figura 4.2.

Las variables ”originales” que contienen la información serían la variables **”V101” y ”V102”**.

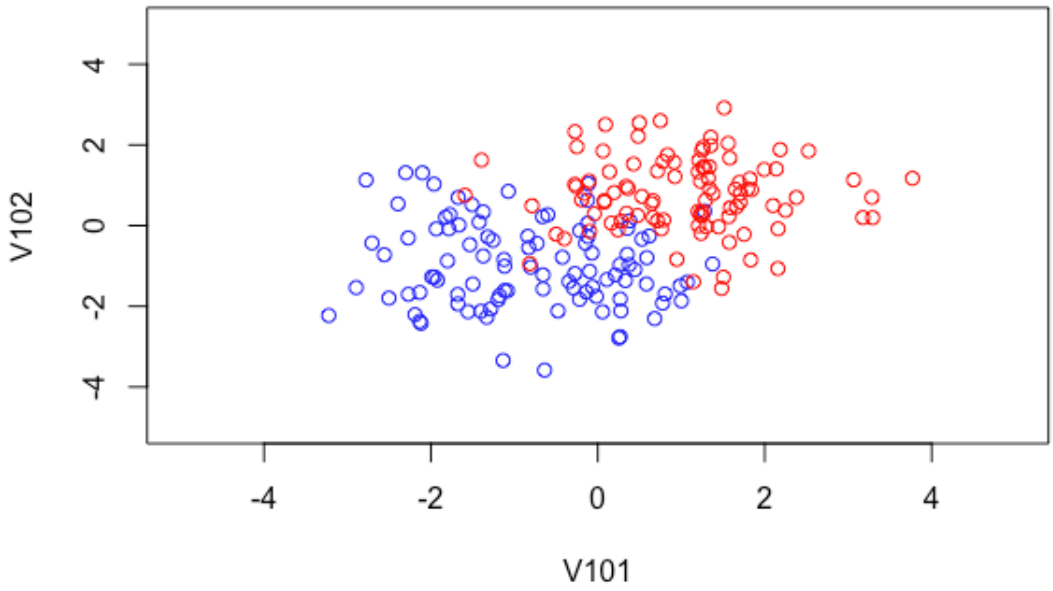


Figura 4.1: Distribución de los grupos en las dos variables.

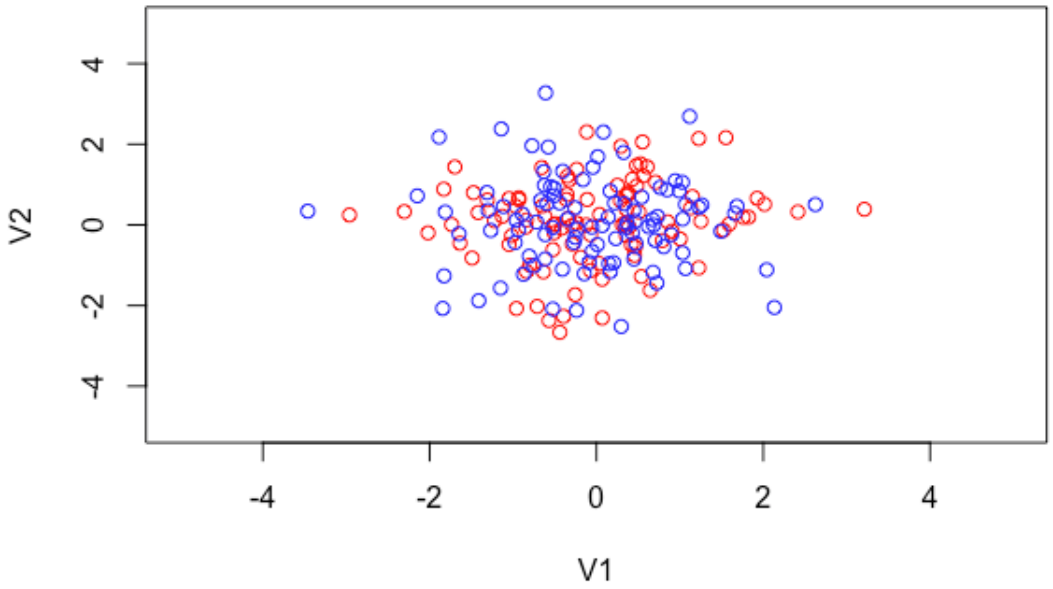


Figura 4.2: Distribución de los grupos en dos variables de ruido.

4.1.1. GLMnet

El algoritmo GLMnet, mencionado en el apartado 2, consiste en aplicar un Modelo Lineal Generalizado para discriminar los grupos y además aplicar una penalización de red elástica para hacer selección de variables. Este algoritmo se ha puesto a prueba gracias a la **librería GLMnet [22] del paquete GLMnet en R**.

Lo primero que habría que hacer sería estimar el α y el λ óptimos para este conjunto de datos,

esto se puede hacer por validación cruzada, como se explicó en el apartado de redes elásticas, la función ”cva.glmnet” del paquete ”glmnetUtils”[23], estiman por *Cross Validation* estos valores, en la figura 4.3 se puede observar cómo el procedimiento GLMnet encuentra las dos variables que separan ambos grupos, la $V101$ y la $V102$ puesto que son las variables a las que da más peso en la regresión para cualquiera de los α, sin embargo, a esas variables se les da poco peso, y no son las únicas seleccionadas para ningún valor de α, nótese que el intercept tiene más peso en la regresión que cualquiera de las variables, por ello la tasa de acierto no supera para ningún alpha el $30\,\%$ (Figura 4.4)

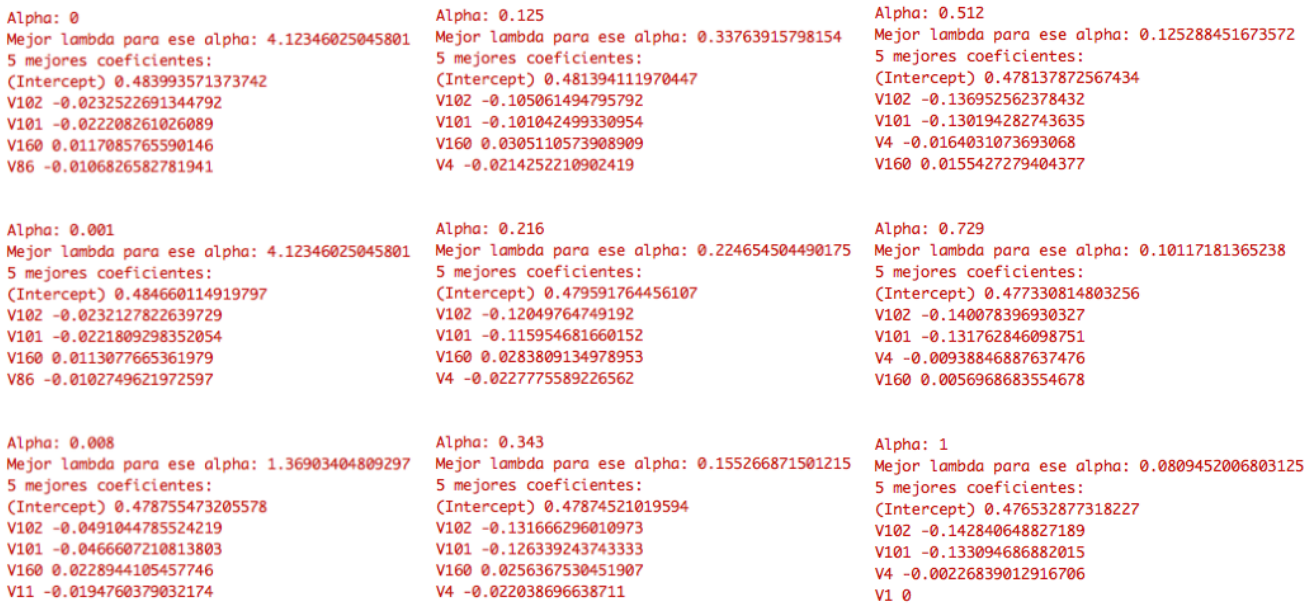


Figura 4.3: Coeficientes estimados por el procedimiento GLMnet

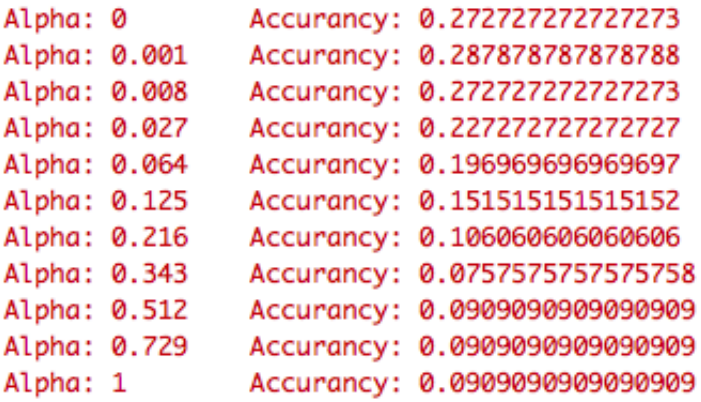


Figura 4.4: Tasas de acierto para GLMnet

4.1.2. Sparse Linear Discrimiant Analysis

El algoritmo Sparse LDA, mencionado ya en el apartado 2.1, consiste en aplicar un Análisis Discriminante Lineal, para posteriormente aplicar, al igual que en el GLMnet, una penalización Lasso para hacer selección de variables. Este algoritmo se ha puesto a prueba gracias a la **Función sda del paquete sparseLDA [24] en R**.

Se le introduce al algoritmo un criterio de parada, que puede ser o por número máximo de iteraciones o por tolerancia, parámetros ya explicados anteriormente, y un número máximo de variables seleccionadas. En este caso se han introducido $2variables$, que deberían coincidir con el número de variables que discriminarían ambos grupos.

En la figura 4.5 se puede apreciar la salida del algoritmo, en la cuál converge en dos iteraciones (En la primera ya encontró el mínimo coste de ridge, y al igual que el método GLMnet, encuentra las variables discriminantes de ambos grupos y realiza un LDA con ellas para formar un discriminante lineal con una tasa de éxito de $92.42\,\%$

```
ite:  1  ridge cost:  48.05043  |b|_1:  0.6035977
ite:  2  ridge cost:  48.05043  |b|_1:  0.6035977
final update, total ridge cost:  48.05043  |b|_1:  0.6035977
          Reference
Prediction Grupo 1 Grupo 2
   Grupo 1      31       4
   Grupo 2       1      30
 Accuracy
0.9242424
[1] "Mejores variables: "
[1] 101 102
```

Figura 4.5: Resultados del Sparse LDA sobre los datos.

4.1.3. Sparse Mixture Discriminant Analysis

El algoritmo Sparse MDA, ya definido en apartados anteriores (2.2.3) consiste en realizar una discriminación por mixturas de normales en lugar de una discriminación únicamente con un par de normales como haría el discriminante lineal, además de esto se le añadiría el método Sparse de selección de variables como en el LDA.

Este algoritmo se ha puesto a prueba gracias a la **Función smda del paquete sparseLDA [24] en R**.

Se le introduce al algoritmo un criterio de parada, que puede ser o por número máximo de iteraciones o por tolerancia, parámetros ya explicados anteriormente, un número máximo de variables seleccionadas, en este caso se han introducido **2 variables**, que deberían coincidir con el número de variables que discriminarían ambos grupos, y el número de sub-clases(Mixturas de normales) que se desea tener por cada clase, en este caso se ha decidido dividir una clase en 5 sub-clases.

Sin embargo, como se puede apreciar en la figura 4.6, aún que en el plano teórico este algoritmo no debería tener problemas para encontrar los dos grupos ya que están bastante separados, en la práctica no parece que se estén cumpliendo las suposiciones, las estimaciones teóricas deberían encontrar correctamente las variables pero el estimador práctico no parece que encuentre esas variables. También puede ser que el algoritmo no esté bien optimizado.

```
          Reference
Prediction Grupo 1 Grupo 2
   Grupo 1      19      17
   Grupo 2      12      18
 Accuracy
0.5606061
[1] "Mejores variables: "
[1]  96 110
```

Figura 4.6: Resultados del Sparse LDA con Mixturas de Normales sobre los datos.

4.1.4. Random Forest Variable Selection

El Random Forest [18] es un método discriminante que consiste en la combinación o ensemble de árboles predictores de tal manera que cada árbol dependa de los valores de un único vector aleatorio independiente de cada árbol.

Random Forest ya realiza implícitamente una selección de variables, sin embargo puede llegar a ser poco precisa e inestable, por ello en esta sección se hará una propuesta de un procedimiento de dos pasos en el cuál se promediarán las mejores variables que detecten varias ejecuciones de Random Forest Anidadas sobre un mismo conjunto de entrenamiento.

Para poner en práctica este algoritmo se ha utilizado la **función VSURF del paquete VSURF [25] de R**.

Introduciéndole a la función un número de árboles para cada modelo RandomForest, el número de variables seleccionadas en cada split (m), y el número de ejecuciones del modelo RandomForest para el conjunto de datos, el algoritmo toma como predeterminado 2000 ejecuciones de random forest.

El tiempo total de ejecución del algoritmo sería el siguiente:

```
39 variables at thresholding step (in 2 mins)
28 variables at interpretation step (in 1.4 mins)
3 variables at prediction step (in 28.2 secs)
```

Las variables que ha seleccionado el algoritmo en las etapas del mismo:

Paso 1. Cálculo del Umbral. Variables seleccionadas:

```
    101 102  35 201 120 175 152 155 134  49  62  76 144  67  13 183 188  56 189 199 125  93
         10 130 176 138   3  64  17  77 135 118  40 173 117 154 5 121 182
```

Paso 2. Interpretación.

```
    101 102  35 201 120 175 152 155 134  49  62  76 144  67  13 183 188  56 189 199 125  93
         10 130 176 138   3  64
```

Paso 3. Predicción.

```
1    101
```

El algoritmo estima que con una única variable se pueden discriminar bien los grupos, lo que es correcto puesto que la matriz de confusión para el conjunto tests en esta situación sería la siguiente:

	Reference	
Prediction	Grupo 1	Grupo 2
Grupo 1	27	4
Grupo 2	7	28

Accuracy
0.8333333

Figura 4.7: Resultados del RF con la variable seleccionada.

También se puede observar en la importancia que le da las variables en uno de las ejecuciones de los modelos Random Forest que las variables $V101$ y $V102$ son las que más importancia tendrían a la hora de clasificar ambos grupos :

```
1      V101      V102       V35      V120      V201      V175      V155
2 8.6764741 6.4393056 1.0922275 0.9248310 0.6295362 0.6052104 0.5608225
3      V144      V183      V173
4 0.5424863 0.5321977 0.5243395
```

Suponiendo un modelo RF con las dos variables $V101$ y $V102$, la precisión aumentaría a un 0.9242, y se podría realizar un gráfico de pertenencia a grupos(4.8) para ambas variables donde se puede observar la regla de clasificación:

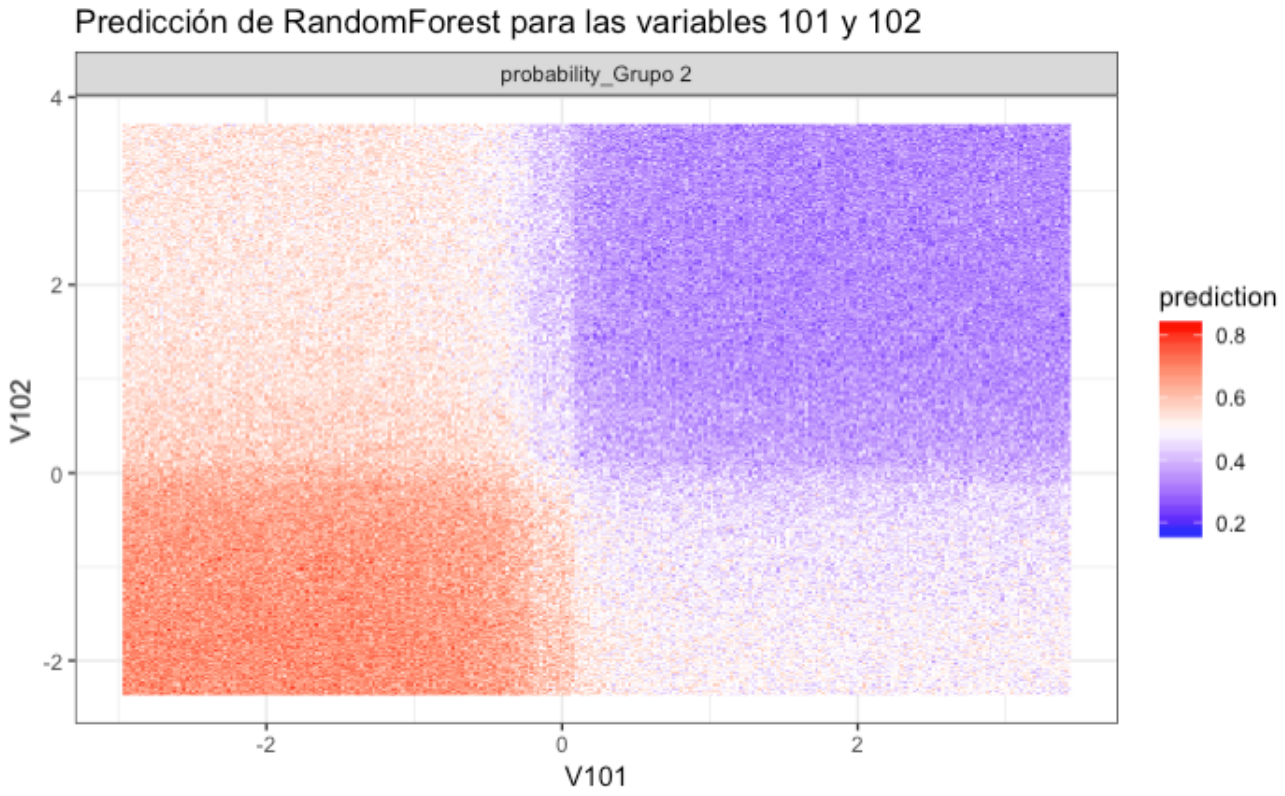


Figura 4.8: Predicciones del modelo RF en función de sus valores para la variable V101 y V102.

4.2. Situación 2: Dos grupos, dos variables con la información y matriz de varianzas-covarianzas perpendicular (+).

Se van a generar dos muestras de normales bivariantes con matrices de varianzas-covarianzas perpendiculares para cada uno de los dos grupos, con esto se obtendrán las dos variables que serán capaces de discriminar ambos grupos, sin embargo, esta es la situación más desfavorable para los métodos Sparse mencionados en el capítulo 2.

100 observaciones de Normales $N(\begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}1 & 0\\0 & 100\end{pmatrix})$ para el **grupo 1**.

100 observaciones de Normales $N(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}100 & 0\\0 & 1\end{pmatrix})$ para el **grupo 2**.

En la figura 4.9 se puede observar la distribución de estos grupos en las dos variables generadas.

Se simulan posteriormente 200 variables aleatorias a mayores para generar ruido en los datos, por ejemplo, distribuciones $N(0, 100)$.

Las variables ”originales” que contienen la información serían la variables ’’**V101” y ”V102”**.

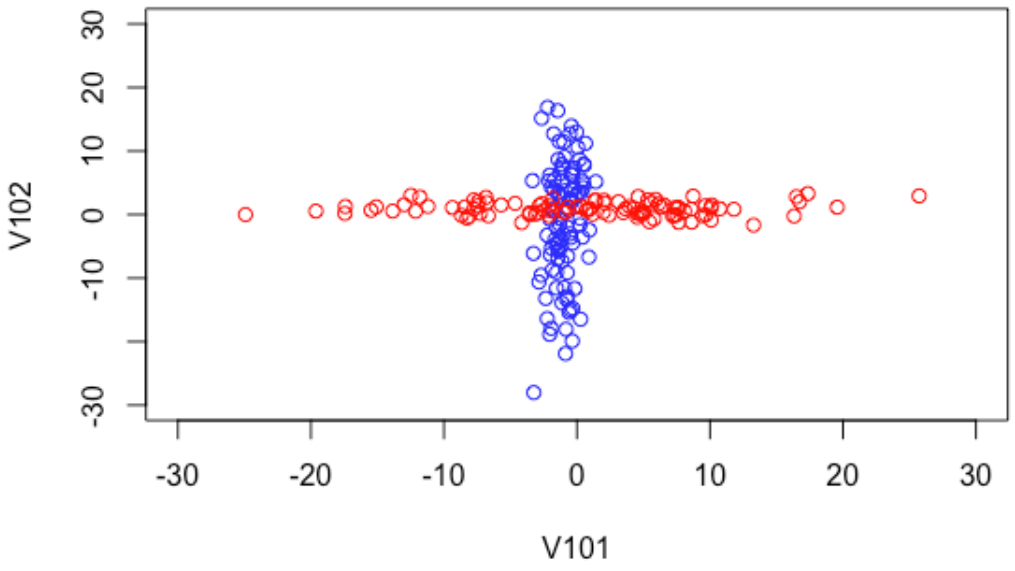


Figura 4.9: Distribución de los grupos en las dos variables.

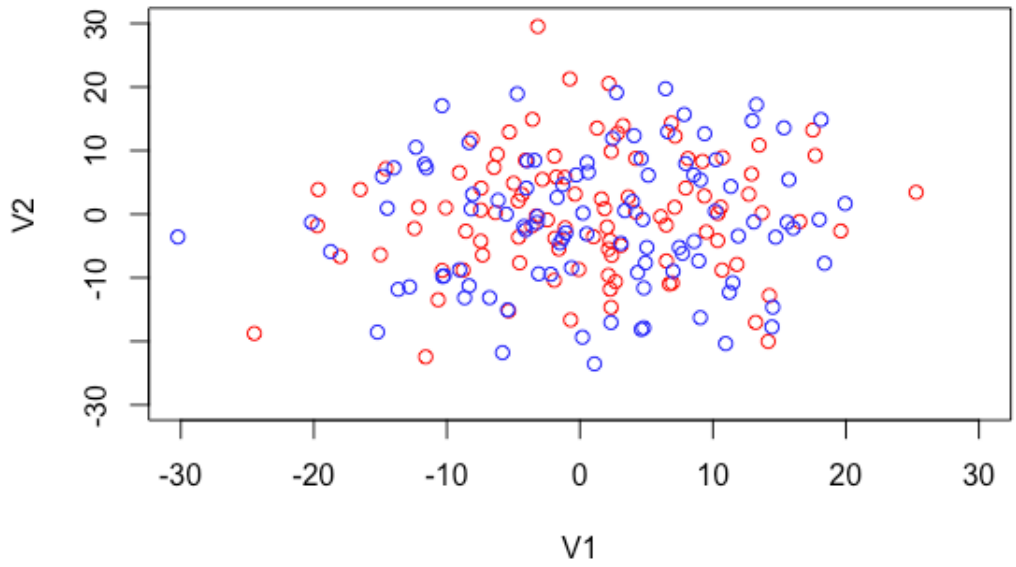


Figura 4.10: Distribución de los grupos en dos variables de ruido.

4.2.1. GLMnet

Lo primero que habría que hacer sería estimar el α y el λ óptimos para este conjunto de datos mediante la función ”cva.glmnet” del paquete ”glmnetUtils”.

Como se puede apreciar en la figura 4.11, GLMnet no es capaz de diferenciar las variables salvo para $\alpha = 0$, y aún para $\alpha = 0$ le da aún menor valor a las variables que en la situación anterior por lo que las predicciones son prácticamente aleatorias como se observa en la figura 4.12

```
Alpha: 0
Mejor lambda para ese alpha: 107.724810861761
5 mejores coeficientes:
(Intercept) 0.507639701590141
V101 -7.42678223746182e-05
V102 -5.8651366304838e-05
V158 4.93073689300666e-05
V165 4.82227527159515e-05

Alpha: 0.125
Mejor lambda para ese alpha: 0.861798486894091
5 mejores coeficientes:
(Intercept) 0.507462686567165
V1 0
V2 0
V3 0
V4 0

Alpha: 0.512
Mejor lambda para ese alpha: 0.210400021214378
5 mejores coeficientes:
(Intercept) 0.507462686567165
V1 0
V2 0
V3 0
V4 0

Alpha: 0.001
Mejor lambda para ese alpha: 107.724810861761
5 mejores coeficientes:
(Intercept) 0.507462686567165
V1 0
V2 0
V3 0
V4 0

Alpha: 0.216
Mejor lambda para ese alpha: 0.498725976211858
5 mejores coeficientes:
(Intercept) 0.507462686567165
V1 0
V2 0
V3 0
V4 0

Alpha: 0.729
Mejor lambda para ese alpha: 0.147770659618328
5 mejores coeficientes:
(Intercept) 0.507462686567165
V1 0
V2 0
V3 0
V4 0

Alpha: 0.008
Mejor lambda para ese alpha: 13.4656013577202
5 mejores coeficientes:
(Intercept) 0.507462686567165
V1 0
V2 0
V3 0
V4 0

Alpha: 0.343
Mejor lambda para ese alpha: 0.314066503970149
5 mejores coeficientes:
(Intercept) 0.507462686567165
V158 9.8043649291408e-19
V1 0
V2 0
V3 0

Alpha: 1
Mejor lambda para ese alpha: 0.107724810861761
5 mejores coeficientes:
(Intercept) 0.507462686567165
V1 0
V2 0
V3 0
V4 0
```

Figura 4.11: Coeficientes estimados por el procedimiento GLMnet

```
Alpha: 0        Accurancy: 0.515151515151515
Alpha: 0.001    Accurancy: 0.515151515151515
Alpha: 0.008    Accurancy: 0.515151515151515
Alpha: 0.027    Accurancy: 0.515151515151515
Alpha: 0.064    Accurancy: 0.515151515151515
Alpha: 0.125    Accurancy: 0.515151515151515
Alpha: 0.216    Accurancy: 0.515151515151515
Alpha: 0.343    Accurancy: 0.515151515151515
Alpha: 0.512    Accurancy: 0.515151515151515
Alpha: 0.729    Accurancy: 0.515151515151515
Alpha: 1        Accurancy: 0.515151515151515
```

Figura 4.12: Tasas de acierto para GLMnet

4.2.2. Sparse Linear Discrimiant Analysis

Se le introduce al algoritmo Sparse LDA un criterio de parada, que puede ser o por número máximo de iteraciones o por tolerancia, parámetros ya explicados anteriormente, y un número máximo de variables seleccionadas. En este caso se han introducido $2variables$, que deberían coincidir con el número de variables que discriminarían ambos grupos.

En la figura 4.13 se puede apreciar la salida del algoritmo, en la cuál converge en dos iteraciones (En la primera ya encontró el mínimo coste de ridge), sin embargo, como se intuía, el algoritmo, no es capaz de encontrar las variables que discriminarían los grupos ya que las matrices de varianzas covarianzas son totalmente perpendiculares

```
ite:  1  ridge cost:  133.6103  |b|_1:  0.000670872
ite:  2  ridge cost:  133.6103  |b|_1:  0.000670872
final update, total ridge cost:  133.6103  |b|_1:  0.000670872
          Reference
Prediction Grupo 1 Grupo 2
   Grupo 1      21      11
   Grupo 2      13      21
 Accuracy
0.6363636
[1] "Mejores variables: "
[1]  49 165
```

Figura 4.13: Resultados del Sparse LDA sobre los datos.

4.2.3. Sparse Mixture Discriminant Analysis

Se le introduce al algoritmo un criterio de parada, que puede ser o por número máximo de iteraciones o por tolerancia, parámetros ya explicados anteriormente, un número máximo de variables seleccionadas, en este caso se han introducido **2 variables**, que deberían coincidir con el número de variables que discriminarían ambos grupos, y el número de sub-clases(Mixturas de normales) que se desea tener por cada clase, en este caso se ha decidido dividir una clase en 5 sub-clases.

Sin embargo, como se puede apreciar en la figura 4.14, aún que en el plano teórico este algoritmo no debería tener problemas para encontrar los dos grupos, en la práctica no parece que se estén cumpliendo las suposiciones. También puede ser que el algoritmo no esté bien optimizado.

```
          Reference
Prediction Grupo 1 Grupo 2
   Grupo 1      16      16
   Grupo 2      19      15
Accuracy
0.469697
[1] "Mejores variables: "
[1]  96 159
```

Figura 4.14: Resultados del Sparse LDA con Mixturas de Normales sobre los datos.

4.2.4. Random Forest Variable Selection

Introduciéndole a la función un número de árboles para cada modelo RandomForest, el número de variables seleccionadas en cada split (m), y el número de ejecuciones del modelo RandomForest para el conjunto de datos, el algoritmo toma como predeterminado 2000 ejecuciones de random forest.

El tiempo total de ejecución del algoritmo sería el siguiente:

```
25 variables at thresholding step (in 2.14 mins)
9 variables at interpretation step (in 41.5 secs)
1 variables at prediction step (in 8.6 secs)
```

Las variables que ha seleccionado el algoritmo en las etapas del mismo:

Paso 1. Cálculo del Umbral. Variables seleccionadas:

```
    101 102 194 158  49 165  79 192  74  36 109 169  19  43  76  42  14 170 117 180  20  96
         15 184 198
```

Paso 2. Interpretación.

```
    101 102 194 158  49 165  79 192  74
```

Paso 3. Predicción.

```
    101
```

El algoritmo estima que con una única variable $V101$ se podría discriminar correctamente las dos variables

```
            Reference
Prediction Grupo 1 Grupo 2
   Grupo 1      27       2
   Grupo 2       7      30
 Accuracy
0.8636364
```

Figura 4.15: Resultados del RF con la variable seleccionada.

También se puede observar en la importancia que le da las variables en uno de las ejecuciones de los modelos Random Forest que las variables $V101$ y $V102$ son las que más importancia tendrían a la hora de clasificar ambos grupos :

```
     V101      V102       V35      V120      V201      V175      V155
8.6764741 6.4393056 1.0922275 0.9248310 0.6295362 0.6052104 0.5608225
     V144      V183      V173
0.5424863 0.5321977 0.5243395
```

Suponiendo un modelo RF con las dos variables $V101$ y $V102$ se podría realizar un gráfico de pertenencia a los grupos(4.16) para ambas variables donde se puede observar la regla de clasificación:

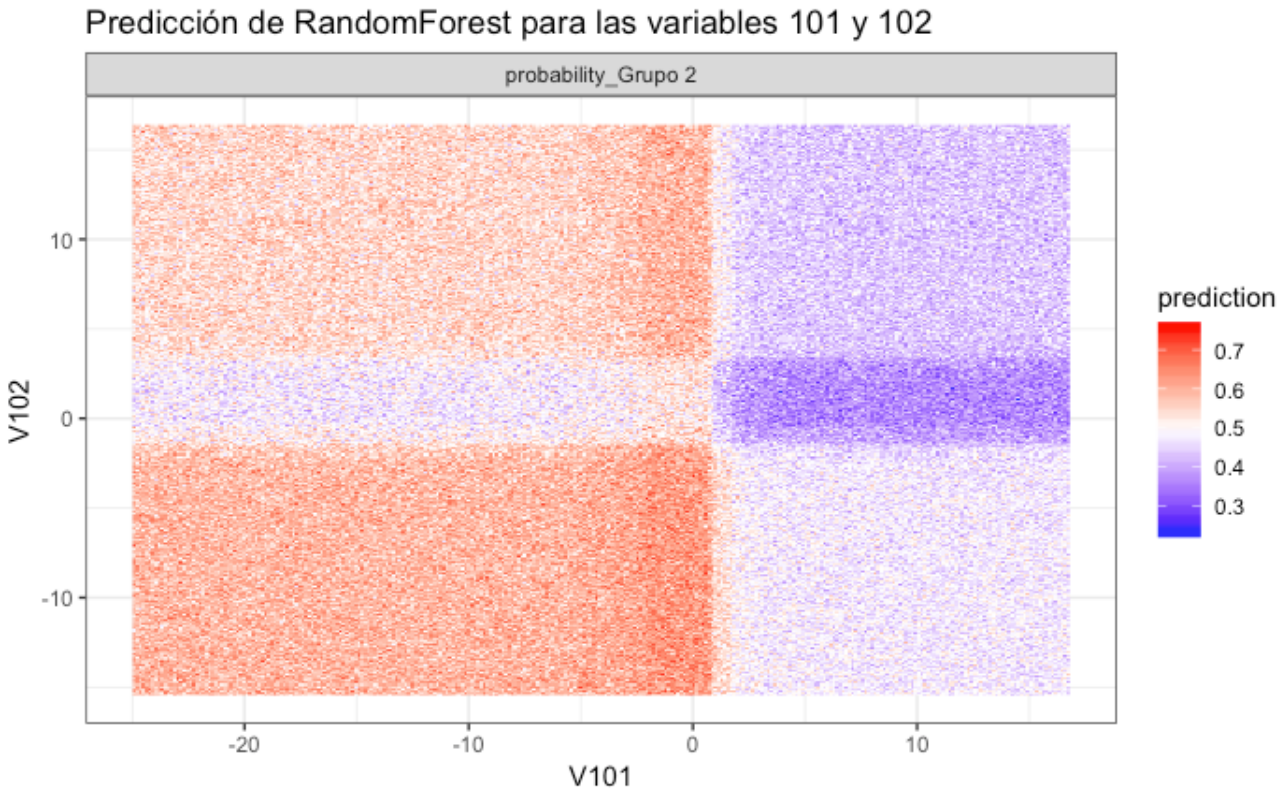


Figura 4.16: Predicciones del modelo RF en función de sus valores para la variable V101 y V102.

4.3. Situación 3: Dos grupos, dos variables con la información y matriz de varianzas-covarianzas perpendicular rotada 45 grados ($\times$).

Se van a generar dos muestras de normales bivariantes con matrices de varianzas-covarianzas perpendiculares para cada uno de los dos grupos, esto implicaría que los métodos Sparse no sean capaces de seleccionar las variables informativas, sin embargo, el método del Random Forest sigue funcionando bien incluso para variables perpendiculares en forma de "$+$", esto podría cambiar si estas variables en vez de formar una cruz, formasen una $\times$ puesto que al estar compuesto de reglas de clasificación por árboles, los cortes de una $\times$ podrían ser demasiado complejos como para que sea capaz de detectarlos.

Para ello se rotarán los valores de las observaciones para esas variables 45 grados mediante la transformación:

$$\begin{pmatrix} V101_1 & V102_1 \\ V101_2 & V102_2 \\ \vdots & \vdots \\ V101_n & V102_n \end{pmatrix} \times \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \tag{4.1}$$

100 observaciones de Normales $N(\begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix})$ para el **grupo 1**.

100 observaciones de Normales $N(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 100 & 0 \\ 0 & 1 \end{pmatrix})$ para el **grupo 2**.

Transformar las variables V101 y V102 rotándolas 45 grados mediante la fórmula 4.1

En la figura 4.17 se puede observar la distribución de estos grupos en las dos variables generadas.

Se simulan posteriormente 200 variables aleatorias a mayores para generar ruido en los datos, por ejemplo, distribuciones $N(0, 100)$.

Las variables "originales" que contienen la información serían la variables **"V101" y "V102"**.

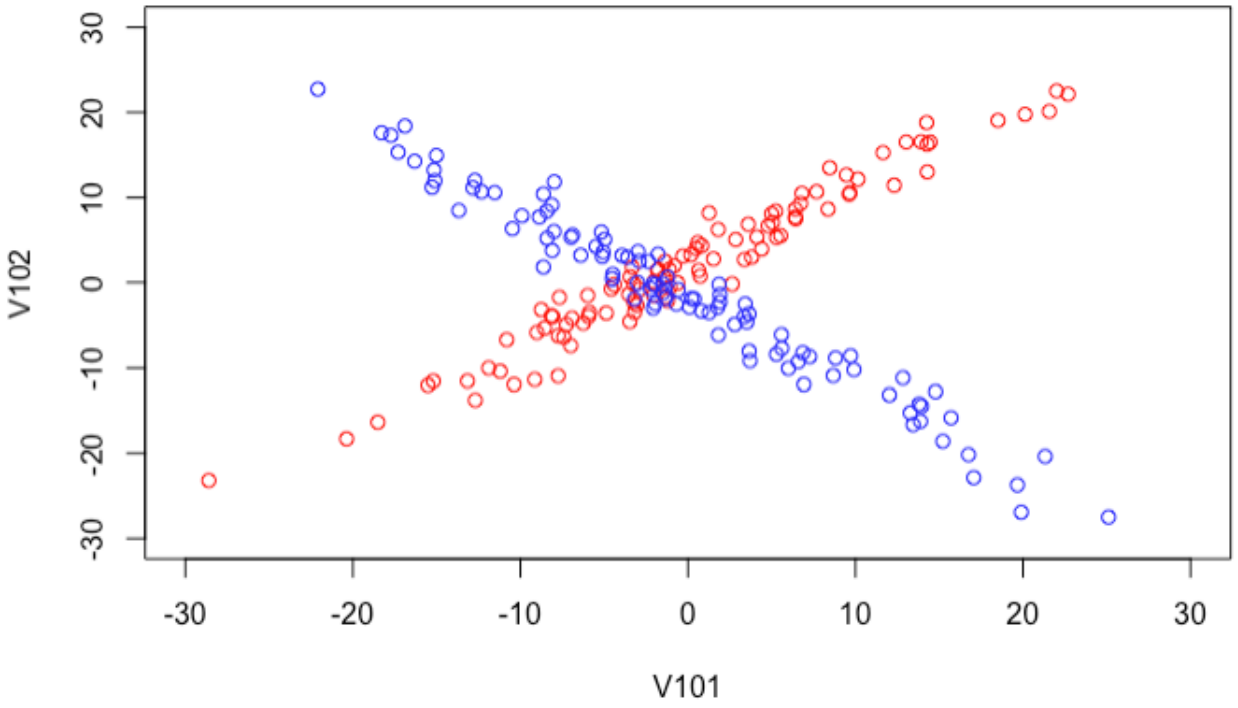


Figura 4.17: Distribución de los grupos en las dos variables.

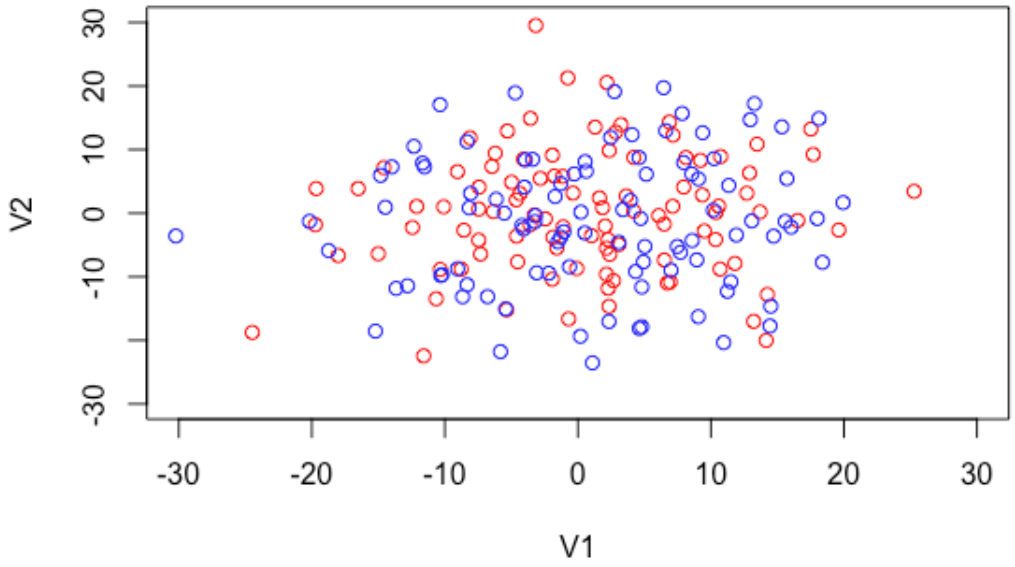


Figura 4.18: Distribución de los grupos en dos variables de ruido.

4.3.1. GLMnet

Lo primero que habría que hacer sería estimar el α y el λ óptimos para este conjunto de datos mediante la función ”cva.glmnet” del paquete ”glmnetUtils”.

Como se puede apreciar en la figura 4.19, GLMnet no es capaz de diferenciar las variables para ninguna de los α, y las predicciones siguen siendo aleatorias completamente 4.20

```
Alpha: 0
Mejor lambda para ese alpha: 109.873939252234
5 mejores coeficientes:
(Intercept) 0.514809110128184
V121 -5.42683989921105e-05
V108 4.5573784565435e-05
V197 -4.13695342951927e-05
V52 -4.06865556087849e-05

Alpha: 0.064
Mejor lambda para ese alpha: 1.71678030081615
5 mejores coeficientes:
(Intercept) 0.514925373134329
V1 0
V2 0
V3 0
V4 0

Alpha: 0.512
Mejor lambda para ese alpha: 0.214597537602019
5 mejores coeficientes:
(Intercept) 0.514925373134329
V1 0
V2 0
V3 0
V4 0

Alpha: 0.001
Mejor lambda para ese alpha: 109.873939252234
5 mejores coeficientes:
(Intercept) 0.514925373134329
V1 0
V2 0
V3 0
V4 0

Alpha: 0.125
Mejor lambda para ese alpha: 0.87899151401787
5 mejores coeficientes:
(Intercept) 0.514925373134329
V1 0
V2 0
V3 0
V4 0

Alpha: 0.729
Mejor lambda para ese alpha: 0.150718709536672
5 mejores coeficientes:
(Intercept) 0.514925373134329
V121 -4.22041159717071e-18
V1 0
V2 0
V3 0

Alpha: 0.008
Mejor lambda para ese alpha: 13.7342424065292
5 mejores coeficientes:
(Intercept) 0.514925373134329
V1 0
V2 0
V3 0
V4 0

Alpha: 0.216
Mejor lambda para ese alpha: 0.508675644686267
5 mejores coeficientes:
(Intercept) 0.514925373134329
V121 -8.46390966775817e-19
V1 0
V2 0
V3 0

Alpha: 1
Mejor lambda para ese alpha: 0.109873939252234
5 mejores coeficientes:
(Intercept) 0.514925373134329
V121 -3.04355252531719e-18
V1 0
V2 0
V3 0
```

Figura 4.19: Coeficientes estimados por el procedimiento GLMnet

```
Alpha: 0        Accurancy: 0.53030303030303
Alpha: 0.001    Accurancy: 0.53030303030303
Alpha: 0.008    Accurancy: 0.53030303030303
Alpha: 0.027    Accurancy: 0.53030303030303
Alpha: 0.064    Accurancy: 0.53030303030303
Alpha: 0.125    Accurancy: 0.53030303030303
Alpha: 0.216    Accurancy: 0.53030303030303
Alpha: 0.343    Accurancy: 0.53030303030303
Alpha: 0.512    Accurancy: 0.53030303030303
Alpha: 0.729    Accurancy: 0.53030303030303
Alpha: 1        Accurancy: 0.53030303030303
```

Figura 4.20: Tasas de acierto para GLMnet

4.3.2. Sparse Linear Discrimiant Analysis

Se le introduce al algoritmo Sparse LDA un criterio de parada, que puede ser o por número máximo de iteraciones o por tolerancia, parámetros ya explicados anteriormente, y un número máximo de variables seleccionadas. En este caso se han introducido $2variables$, que deberían coincidir con el número de variables que discriminarían ambos grupos.

En la figura 4.21 se puede apreciar la salida del algoritmo, en la cuál converge en dos iteraciones (En la primera ya encontró el mínimo coste de ridge), sin embargo, sigue funcionando mal el algoritmo puesto que aún que estén rotadas las variables, siguen siendo perpendiculares por lo que un método Sparse no será capaz de seleccionar variable.

```
ite:  1  ridge cost:  132.2045  |b|_1:  0.003598999
ite:  2  ridge cost:  132.2045  |b|_1:  0.003598999
final update, total ridge cost:  132.2045  |b|_1:  0.003598999
          Reference
Prediction Grupo 1 Grupo 2
   Grupo 1      17      18
   Grupo 2      18      13
 Accuracy
0.4545455
[1] "Mejores variables: "
[1]  23 121
```

Figura 4.21: Resultados del Sparse LDA sobre los datos.

4.3.3. Sparse Mixture Discriminant Analysis

Se le introduce al algoritmo un criterio de parada, que puede ser o por número máximo de iteraciones o por tolerancia, parámetros ya explicados anteriormente, un número máximo de variables seleccionadas, en este caso se han introducido **2 variables**, que deberían coincidir con el número de variables que discriminarían ambos grupos, y el número de sub-clases(Mixturas de normales) que se desea tener por cada clase, en este caso se ha decidido dividir una clase en 5 sub-clases.

Sin embargo, como se puede apreciar en la figura 4.14, aún que en el plano teórico este algoritmo no debería tener problemas para encontrar los dos grupos, en la práctica no parece que se estén cumpliendo las suposiciones. También puede ser que el algoritmo no esté bien optimizado.

```
          Reference
Prediction Grupo 1 Grupo 2
   Grupo 1       9       4
   Grupo 2      26      27
 Accuracy
0.5454545
[1] "Mejores variables: "
[1]  45 116
```

Figura 4.22: Resultados del Sparse LDA con Mixturas de Normales sobre los datos.

4.3.4. Random Forest Variable Selection

Introduciéndole a la función un número de árboles para cada modelo RandomForest, el número de variables seleccionadas en cada split (m), y el número de ejecuciones del modelo RandomForest para el conjunto de datos, el algoritmo toma como predeterminado 2000 ejecuciones de random forest.

El tiempo total de ejecución del algoritmo sería el siguiente:

```
1 26 variables at thresholding step (in 2.23 mins)
2 8 variables at interpretation step (in 54.12 secs)
3 1 variables at prediction step (in 8.89 secs)
```

Las variables que ha seleccionado el algoritmo en las etapas del mismo:

Paso 1. Cálculo del Umbral. Variables seleccionadas:

```
    121  60 145 113  23 102 200  87  45  38 193 117 177 101  78 141 148 150 115  83  54 109
        166  75  62 182
```

Paso 2. Interpretación.

```
    121 60  145 113  23 102 200  87
```

Paso 3. Predicción.

```
    121
```

El algoritmo estima que con una única variable $V121$ se podría discriminar correctamente las dos variables, sin embargo esto no es así puesto que esa variable es una de las generadas como ruido, al igual que la siguiente variable que el RandomForest ha considerado como relevante para el modelo, la $V60$

Prediction	Grupo 1	Grupo 2
Grupo 1	7	17
Grupo 2	28	14

Accuracy
0.3181818

Figura 4.23: Resultados del RF con la variable seleccionada.

También se puede observar en la importancia que le da las variables en uno de las ejecuciones de los modelos Random Forest que las variables $V121$ y $V60$ son las que más importancia tendrían a la hora de clasificar ambos grupos, pero esa importancia es demasiado baja y similar a otras variables generadas aleatoriamente como ruido. :

```
     V121       V60      V145      V113       V45      V200       V87
1.1340038 0.9756581 0.8284575 0.7105893 0.7103893 0.6556729 0.6090184
      V54      V109       V23
0.5942074 0.5939799 0.5859525
```

Suponiendo un modelo RF con las dos variables $V121$ y $V60$ se podría realizar un gráfico de pertenencia a los grupos(4.24) para ambas variables donde se puede observar la regla de clasificación, totalmente aleatoria, sin poder distinguir ningún patrón:

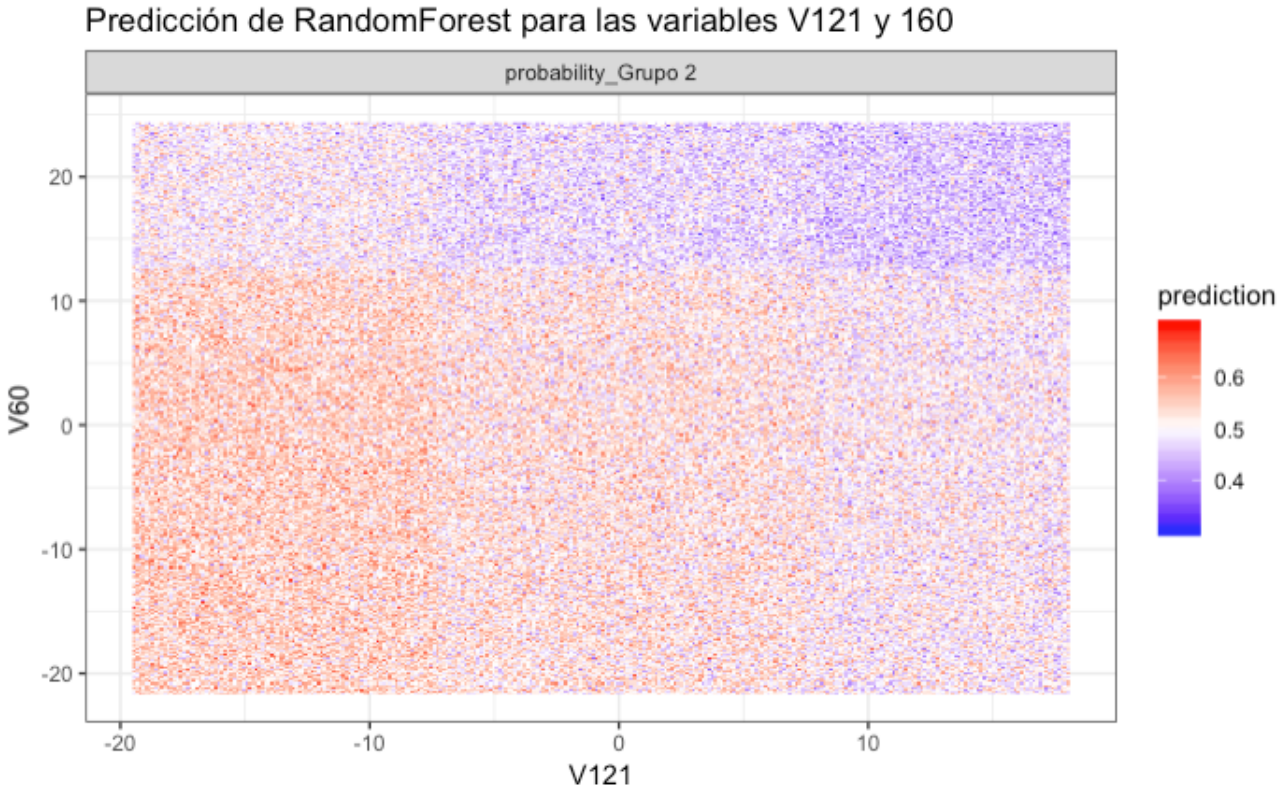


Figura 4.24: Predicciones del modelo RF en función de sus valores para la variable V121 y V60.

4.4. Situación 4: 5 grupos, Cinco variables con la información y matrices de varianzas-covarianzas diferentes para cada variable.

Se van a generar cinco muestras de normales pentavariantes con matrices de varianzas-covarianzas distintas para cada uno de los cinco grupos, esto implicaría que los métodos Sparse no funcionen a la perfección, pero que sean capaces de encontrar alguna de las variables con la información:

100 observaciones de Normales $N(\begin{pmatrix}-8\\0\\0\\0\\0\end{pmatrix}, \begin{pmatrix}5&0&0&0&0\\0&3&0&0&0\\0&0&9&0&0\\0&0&0&4&0\\0&0&0&0&8\end{pmatrix})$ para el **grupo 1**.

100 observaciones de Normales $N(\begin{pmatrix}0\\-4\\0\\0\\0\end{pmatrix}, \begin{pmatrix}8&0&0&0&0\\0&3&0&0&0\\0&0&9&0&0\\0&0&0&4&0\\0&0&0&0&5\end{pmatrix})$ para el **grupo 2**.

100 observaciones de Normales $N(\begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix}, \begin{pmatrix}4&0&0&0&0\\0&8&0&0&0\\0&0&9&0&0\\0&0&0&5&0\\0&0&0&0&3\end{pmatrix})$ para el **grupo 3**.

100 observaciones de Normales $N(\begin{pmatrix}0\\0\\0\\4\\0\end{pmatrix}, \begin{pmatrix}9&0&0&0&0\\0&3&0&0&0\\0&0&8&0&0\\0&0&0&4&0\\0&0&0&0&5\end{pmatrix})$ para el **grupo 4**.

100 observaciones de Normales $N(\begin{pmatrix}0\\0\\0\\0\\8\end{pmatrix}, \begin{pmatrix}3&0&0&0&0\\0&5&0&0&0\\0&0&3&0&0\\0&0&0&3&0\\0&0&0&0&8\end{pmatrix})$ para el **grupo 5**.

En la figura 4.25 se puede observar la distribución de estos grupos en las dos variables generadas.

Se simulan posteriormente 500 variables aleatorias a mayores para generar ruido en los datos, por ejemplo, distribuciones $N(0, 10)$.

Las variables ”originales” que contienen la información serían la variables **”V501”, ”V502”, ”V503”, ”V504”, ”V505”**, aún que, como se verá en las ejecuciones de los algoritmos, la información que proporciona la variable ”V503” será redundante con la variabilidad que explica el resto de las variables

y por lo tanto no será considerada en sus respectivos modelos.

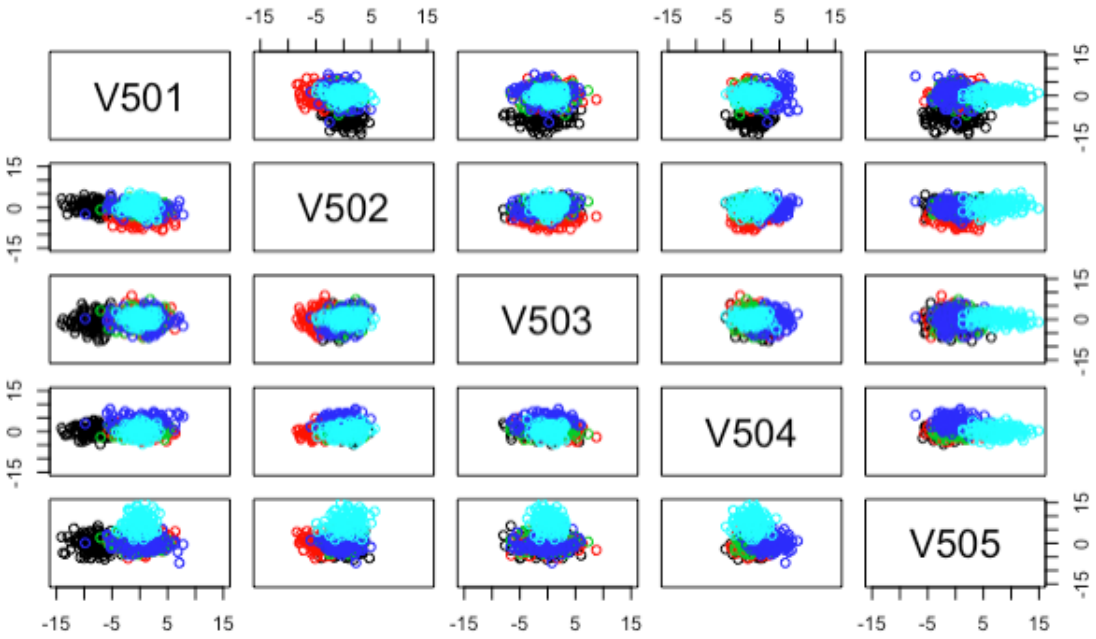


Figura 4.25: Distribución de los grupos en las dos variables.

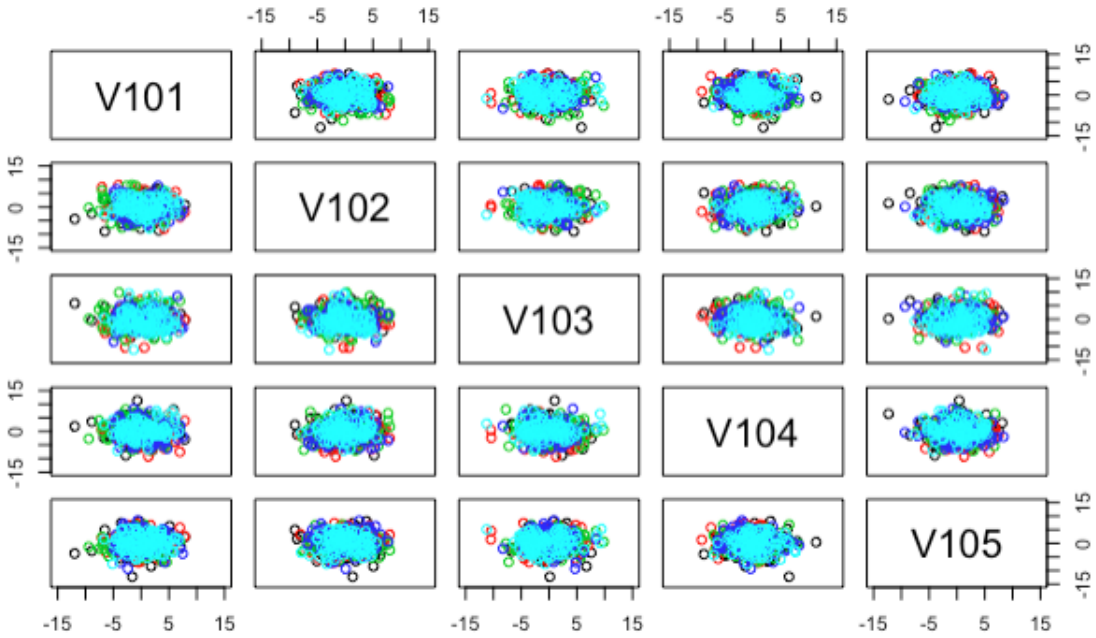


Figura 4.26: Distribución de los grupos en cinco variables de ruido.

4.4.1. GLMnet

Lo primero que habría que hacer sería estimar el α y el λ óptimos para este conjunto de datos mediante la función ”cva.glmnet” del paquete ”glmnetUtils”.

Como se puede apreciar en la figura 4.19, GLMnet si que es capaz de encontrar las variables que discriminarían los grupos para algunos de los α, sin embargo, al darle mucho peso al intercept y demasiado poco a las variables, el algoritmo no será capaz de discriminar los grupos.

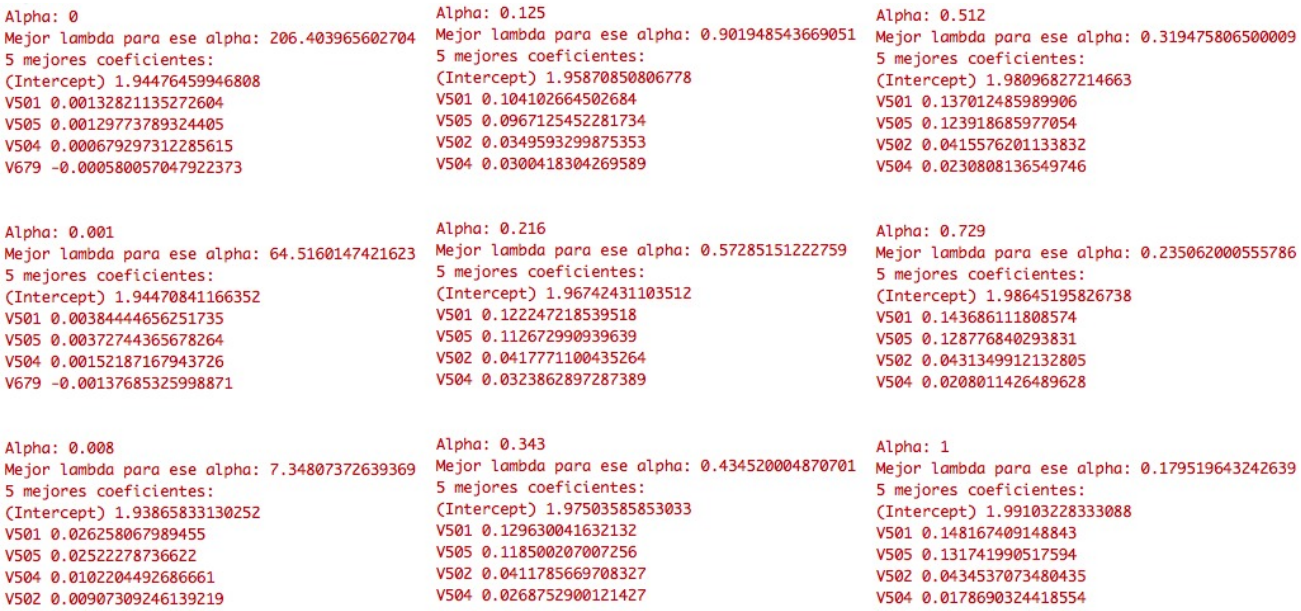


Figura 4.27: Coeficientes estimados por el procedimiento GLMnet

4.4.2. Sparse Linear Discrimiant Analysis

Se le introduce al algoritmo Sparse LDA un criterio de parada, que puede ser o por número máximo de iteraciones o por tolerancia, parámetros ya explicados anteriormente, y un número máximo de variables seleccionadas. En este caso se han introducido $5variables$, que deberían coincidir con el número de variables que discriminarían ambos grupos, en este caso deberían salir 4 variables, **”V501”, ”V502”, ”V504”, ”V505”** ya que la información de la ”V503” ya se encuentra explicada por las otras variables.

En la figura 4.28 se puede apreciar la salida del algoritmo, el algoritmo no funciona del todo bien, sólo encuentra las variables ”V501” y ”V505” que no son suficientes para discriminar los dos grupos.

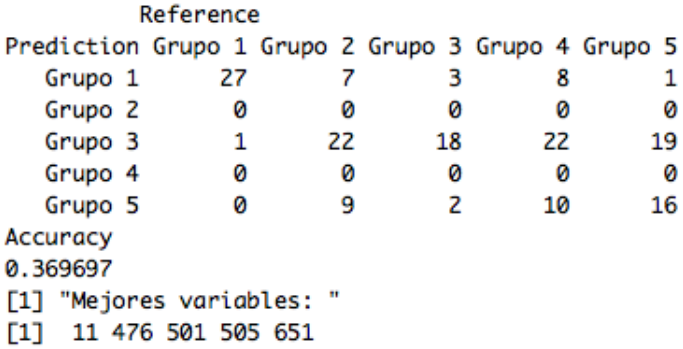


Figura 4.28: Resultados del Sparse LDA sobre los datos.

4.4.3. Sparse Mixture Discriminant Analysis

Se le introduce al algoritmo un criterio de parada, que puede ser o por número máximo de iteraciones o por tolerancia, parámetros ya explicados anteriormente, un número máximo de variables seleccionadas, en este caso se han introducido **5 variables**, que deberían coincidir con el número de variables que discriminarían ambos grupos, y el número de sub-clases(Mixturas de normales) que se desea tener por cada clase, en este caso se ha decidido dividir una clase en 5 sub-clases.

En la figura 4.29 se puede apreciar la salida del algoritmo en la cual el algoritmo de Mixturas si encuentra las variables importantes para el modelo **"V501", "V502", "V504", "V505"**, (La tabla de confusión no es importante en este punto puesto que los datos pueden no ser separables linealmente, lo importante son las variables que el algoritmo ha detectado como las mejores)

```
          Reference
Prediction Grupo 1 Grupo 2 Grupo 3 Grupo 4 Grupo 5
   Grupo 1      23      12       4       9      27
   Grupo 2       0       0       0       0       0
   Grupo 3       1      19      11      19       3
   Grupo 4       4       7       8      12       6
   Grupo 5       0       0       0       0       0
 Accuracy
0.2787879
[1] "Mejores variables: "
[1] 501 502 504 505 640
```

Figura 4.29: Resultados del Sparse LDA con Mixturas sobre los datos.

4.4.4. Random Forest Variable Selection

Introduciéndole a la función diez número de árboles para cada modelo RandomForest, el número de variables seleccionadas en cada split (m), y el número de ejecuciones del modelo RandomForest para el conjunto de datos, en este caso se introducen 10 ejecuciones en lugar de las 2000 que toma como predetermiadas el algoritmo porque es un modelo muy complejo con más de 1000 variables y 500 observaciones.

El tiempo total de ejecución del algoritmo sería el siguiente:

```
4 variables at thresholding step (in 13.70 secs)
4 variables at interpretation step (in 14.69 secs)
4 variables at prediction step (in 0.0014 secs)
```

Las variables que ha seleccionado el algoritmo en las etapas del mismo:

Paso 1. Cálculo del Umbral. Variables seleccionadas:

```
    501 502 504 505
```

Paso 2. Interpretación.

```
1    501 502 504 505
```

Paso 3. Predicción.

```
1    501 502 504 505
```

El algoritmo encuentra desde el paso de ”Tresholding” las cuatro variables influyentes, la matriz de confusión para el modelo RandomForest dando relevancia a esas variables sería la que se muestra en 4.30, no es una buena discriminación, tampoco es relevante para nuestro problema, ya que lo importante es que haya encontrado las variables relevantes frente a todas las variables de ruido:

	Reference				
Prediction	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5
Grupo 1	24	1	2	5	1
Grupo 2	1	21	2	0	0
Grupo 3	3	16	19	18	5
Grupo 4	0	0	0	17	0
Grupo 5	0	0	0	0	30

Accuracy
0.6727273

Figura 4.30: Resultados del RF con la variable seleccionada.

También se puede observar la importancia que le da las variables en una de las ejecuciones de los modelos Random Forest que las variables $V501$, $V505$, $V502$ y $V504$ son las que más relevancia tendrían a la hora de clasificar ambos grupos, además bastante lejos del resto de variables :

```
1       V501        V505        V502        V504        V602         V81         V11
2 10.6145764  8.4728514  6.2383973  4.4520113  0.5574212  0.5544735  0.5408413
3       V934          V4        V882
4  0.5183223  0.5173178  0.5077187
```

Todo el código utilizado en la entrega de este proyecto se encuentra el el fichero ”Métodos de selección de variables enclustering y análisis discriminante.zip”

Capítulo 5

Conclusiones

En esta capítulo se recogen las ideas resultantes del desarrollo de este proyecto y las líneas de trabajo futuro que este podría desencadenar. En primer lugar, con el desarrollo de este proyecto se ha llegado a la conclusión de que existen una gran cantidad de métodos de selección de variables basados en técnicas estadísticas muy eficientes y robustas que, sin embargo, si el número de variables se incrementa hasta el punto de ser mayor que el número de observaciones pueden funcionar peor que una discriminación puramente aleatoria, y que gracias a algoritmos estadísticos de selección de variables que reduzcan la dimensionalidad de los datos para quedarse sólo con variables de vital importancia para el modelo, las técnicas discriminantes sigan teniendo esa eficiencia descrita anteriormente.

En segundo lugar, tras poner a competir los novedosos algoritmos de selección de variables descritos en esta memoria, *Redes elásticas aplicadas a modelos lineales generalizados*, *Sparse Lineal Discriminant Analysis*, *Sparse Mixture Discriminant Analysis* y *Random forest* para la selección de variables, se ha llegado a la conclusión de que los *métodos Sparse* son muy potentes en la selección de variables, ya que aún que el método de clasificación no sea capaz de discriminar los grupos, estos métodos de selección de variables si son capaces de detectar las variables influyentes con capacidad de discriminación en los grupos cuando la variabilidad de las distribuciones de estos es la misma, sin embargo, cuando la variabilidad en ambos grupos es perpendicular, se ha visto como estos métodos no son del todo eficiente ya que no siempre encuentran esas variables influyentes. Para subsanar este problema se propuso el método *Sparse Mixture Discriminant Analysis* que teóricamente si se dividían los grupos con distinta variabilidad en sub-grupos que si tengan la misma variabilidad y se aplicara un método Sparse de selección de variables se debería poder encontrar las variables discriminantes, sin embargo, cuando se ha llevado este algoritmo a la práctica, no ha funcionado correctamente, o la implementación del mismo no está lo suficientemente optimizada. Para esta situación de grupos perpendiculares se propuso el método *Random Forest* que si fue capaz de encontrar las variables discriminantes correctamente puesto que al estar en forma de cruz (+), los cortes que podrían hacer los árboles de decisión serían triviales para ese problema. Para aumentar la complejidad y poner a prueba a este método se propuso rotar 45 grados los grupos de tal manera que quedasen en forma de $\times$ y los cortes que pudiesen hacer los árboles de división no fuesen tan triviales. En esta situación el método de Random Forest no fue capaz de localizar las variables.

Y en tercer y último lugar, se ha determinado por lo tanto que, el algoritmo Random Forest es el que mejor rendimiento ha tenido en las múltiples situaciones que se han propuesto en este proyecto, aún que se le haya encontrado una situación límite en la cuál no ha sido capaz de ubicar las variables informativas y que los métodos Sparse se han desenvuelto muy bien en la mayoría de las situaciones salvo en los casos límite cuando la variabilidad entre los grupos es completamente perpendicular entre ellos y por tanto, sus matrices de varianzas-covarianza son lo más distintas posibles.

5.1. Trabajo futuro

A raíz de las conclusiones obtenidas tras el transcurso del proyecto, han surgido nuevas ideas y lineas de investigación que podrían ser interesantes de cara a un trabajo futuro:

1. Optimizar la implementación práctica en R del algoritmo *Sparse Mixture Discriminant Analysis*
2. Búsqueda de un algoritmo de selección de variables que ubique las variables influyentes si están perpendiculares en forma de $\times$
3. Optimización del método de búsqueda del mejor alpha y el mejor lambda para una red elástica.
4. Optimización del Random Forest para la selección de variables para alcanzar un menor tiempo de ejecución, y añadir un número parámetro de número mínimo de variables a utilizar.

Bibliografía

[1] Análisis discriminante lineal wikipedia. [Online]. Available: https://es.wikipedia.org/wiki/An%C3%A1lisis_discriminante_lineal

[2] Feature selection wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Feature_selection

[3] Lenguaje de progrmació estadístico r. [Online]. Available: https://www.r-project.org

[4] Técnicas de análisis discriminante. [Online]. Available: http://www.estadistica.net/Master-Econometria/Analisis_Discriminante.pdf

[5] Introducción al análisis discriminante. [Online]. Available: http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/AMult/tema6am.pdf

[6] O. Veksler, *Pattern Recognition*. Western Science University, Unknown. [Online]. Available: http://www.csd.uwo.ca/~olga/Courses/CS434a_541a/Lecture8.pdf

[7] Linear discriminant analysis (fisher). [Online]. Available: https://en.wikipedia.org/wiki/Linear_discriminant_analysis

[8] Linear discriminant analysis (bayes). [Online]. Available: https://en.wikipedia.org/wiki/Linear_discriminant_analysis

[9] Quadratic discriminant analysis. [Online]. Available: https://scikit-learn.org/stable/modules/lda_qda.html

[10] Métods de selección de variables. [Online]. Available: http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_1263.pdf

[11] M. W. Trevor Hastie, Robert Tibshirani, *Statitical Learning with Sparsity*. CRC Press, Unknown. [Online]. Available: https://web.stanford.edu/~hastie/StatLearnSparsity_files/SLS_corrected_1.4.16.pdf

[12] Métodos de penalización para la selección de variables. [Online]. Available: http://www.iesta.edu.uy/wp-content/uploads/2014/05/CursoPosgrado_Aprendizaje_Automatico_SCastro_2013.pdf

[13] Glm con penalización de red elaástica. [Online]. Available: http://statweb.stanford.edu/~jhf/ftp/glmnet.pdf

[14] Sparse discriminant analysis. [Online]. Available: https://web.stanford.edu/~hastie/Papers/sda_resubm_daniela-final.pdf

[15] Mixture discriminant analysis. [Online]. Available: http://www.personal.psu.edu/jol2/course/stat597e/notes2/mda.pdf

[16] Em method for mixture discriminant analysis. [Online]. Available: https://core.ac.uk/download/pdf/82197215.pdf

[17] Sparse mixture discriminant analysis. [Online]. Available: https://web.stanford.edu/~hastie/Papers/sda_resubm_daniela-final.pdf

[18] Random forest wikipedia. [Online]. Available: https://es.wikipedia.org/wiki/Random_forest

[19] Bagging. [Online]. Available: https://en.wikipedia.org/wiki/Bagging

[20] Random subspace. [Online]. Available: https://en.wikipedia.org/wiki/Random_subspace_method

[21] Out of bag error. [Online]. Available: https://stackoverflow.com/questions/18541923/what-is-out-of-bag-error-in-random-forests

[22] Glmnet implmentation on r. [Online]. Available: https://cran.r-project.org/web/packages/glmnet/glmnet.pdf

[23] Glmnetutils library on r. [Online]. Available: https://cran.r-project.org/package=glmnetUtils

[24] Sparse lda implementation on r. [Online]. Available: https://cran.r-project.org/package=sparseLDA

[25] Variable selection using random forests implementation on r. [Online]. Available: https://cran.r-project.org/web/packages/VSURF/index.html