FACULTAD DE MATEMÁTICAS
Departamento de Estadística e Investigación Operativa

Regresión Lasso

Presentado por:
Laura Ramos Castillo

Supervisado por:
Dr. Rafael Blanquero Bravo
Dr. Emilio Carrizosa Priego

Junio 2018

Índice general

Abstract **5**

1. Introducción **7**

2. El modelo lineal **9**

3. Medidas de bondad de ajuste **11**

4. El Lasso **13**
4.1. Formulación como un problema de optimización 13
4.2. Predicción y estimación del parámetro t 16
4.3. Algoritmos para encontrar soluciones 16

5. Elastic Net **19**
5.1. Definición 19
5.2. Solución 20
5.3. Deficiencia del Elastic Net de Naïve 21
5.4. Estimación de Elastic Net 22

6. El Lars **25**
6.1. El Algortimo Lars 25
6.2. El paquete Lars en R 27

7. Ejemplos numéricos **31**
7.1. Primer ejemplo numérico con la base de datos **Prostata** 31
7.1.1. Descripción del paquete de datos 31
7.1.2. Ajuste por mínimos cuadrados (OLS) 31
7.1.3. Ajuste mediante LASSO 37
7.1.4. Stepwise Regresion 41
7.1.5. Elastic Net 42
7.2. Segundo ejemplo numérico con la base de datos **mtcars** 43
7.2.1. Descripción del paquete de datos 43
7.2.2. Ajuste por mínimos cuadrados (OLS) 46
7.2.3. Ajuste mediante LASSO 50
7.2.4. Stepwise Regresion 56
7.2.5. Elastic Net 56

Bibliografía **57**

Abstract

Currently we find regression problems in many branches of science, so, as better is the model we use to select variables, better will be solved the problem. The models seek: precise predictions, stability and interpretability.

Traditional methods such as stepwise regression, all subsets regression or ridge regression fail in any of the required requirements. In this text we present the LASSO method (least absolute shrinkage and selection operator), which generally improves stability and predictions. However, LASSO has some limitations that will be solved with Elastic Net.

This work begins with an introduction, motivating, as in this fragment, the purpose and usefulness of this text, then, to refresh the memory will be a concise reminder of the linear model. In order to facilitate the reader's understanding, some measures of goodness of fit will be presented.

Then we present the method mentioned above, LASSO, a formulation of it as an optimization problem, and a way of solving it are presented. In order to solve the limitations of the LASSO, we present the Naïve Elastic Net. Next, we introduce the LARS method, which will provide an optimal implementation of LASSO in R, we provide the reader a summary of the functions that constitute the package lars. To finalize, and to fix ideas, we will make use of two numerical examples implemented in R, in which the solutions obtained with least squares, LASSO, stepwise and Elastic Net will be compared.

Capítulo 1

Introducción

La frase que podría definir el fundamento de este trabajo podría ser "I've got all these variables, but I don't know wich ones to use", "Tengo todas estas variables, pero no se cuáles usar", frase con la que comienza [5].
Nuestro objetivo, por tanto, es encontrar un modelo que nos ayude a tomar esas decisiones, es decir, a decidir qué variables son interesantes y cuáles no lo son.
Problemas de regresión con un gran número de predictores candidatos aparecen en varios campos de la ciencia. Este fenómeno cada vez ocurre con más frecuencia debido a los avances de la tecnología.

Algunos de los requisitos deseados en un modelo de selección de variables son:

- Predicciones precisas
- Modelos interpretables
- Estabilidad, es decir: pequeños cambios en los datos no deberían provocar grandes cambios en los predictores usados.

Los métodos tradicionales de selección de variables, como *stepwise regression*, *all subsets regression* o *ridge regression*, fallan en uno o más de los requisitos anteriores. Procedimientos modernos como *boosting* (Freud and Schapire,1997), *forward stagewise regression* (Hastie et al., 2001) y *LASSO* (Tibshirani, 1996), método del que hablaremos en la sección 4, mejoran generalmente la estabilidad y las predicciones.
Efron et al. (2004) muestra que hay fuertes conexiones entre estos métodos modernos, un método que llaman *least angle regression*, y desarrolla un marco algorítmico que incluye todos estos métodos y proporciona una implementación rápida, que denomina *LARS*. Profundizaremos más en el *LARS* en la sección 6.
Aunque *LASSO* funciona con éxito en muchas ocasiones, tiene algunas limitaciones que veremos en la sección 5, presentaremos en dicha sección el modelo conocido como *Elastic Net*, con el objetivo de solventar dichas limitaciones.

Capítulo 2

El modelo lineal

Consideremos el modelo lineal, expresado como:

$$y = \alpha + \mathbf{X}^t\beta + \epsilon \tag{2.1}$$

donde:

- $\mathbf{X}$ es el vector de variables independientes
- β es un parámetro p dimensional desconocido
- $\alpha \in \mathbb{R}$ es desconocido (término independiente)
- ϵ es un término de error con esperanza $E(\epsilon) = 0$ y varianza $var(\epsilon) = \sigma^2$

A partir de un conjunto de pares (x_1,y_1),..,(x_N,y_N), el método de mínimos cuadrados, en inglés ordinary least squares (OLS) consiste en determinar α, β minimizando la suma de los cuadrados de los errores entre los valores obtenidos y_j y las predicciones $\hat{y}_j = \mathbf{x}_j^t\beta + \alpha$

En otras palabras, el estimador OLS $\hat{\beta}$ se obtiene resolviendo el problema de optimización

$$\min_{\alpha\in\mathbb{R},\beta\in\mathbb{R}^p} \sum_{i=1}^{N}(y_i - \alpha - \mathbf{x}_i^t\beta)^2 \tag{2.2}$$

Si definimos la matriz $X \in \mathbb{R}^{N\times p}$ como:

$$X = \begin{bmatrix} 1 & \mathbf{x}_1 \\ \vdots & \vdots \\ 1 & \mathbf{x}_N \end{bmatrix}$$

Y el vector respuestas

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

El problema (2.2) en forma matricial sería:

$$\min_{\beta \in \mathbb{R}^p} \left(y - X^t \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right)^t \left(y - X^t \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right)$$

Condición necesaria y suficiente de optimalidad

Si llamamos $f(\alpha, \beta)$ a la función que queremos minimizar, la función f es convexa y diferenciable. Por lo tanto una condición necesaria y suficiente para que (α, β) sea solución óptima de (2.2) es:

$$\nabla f(\alpha, \beta) = 0 \tag{2.3}$$

Bajo la hipótesis de que $X^t X$ es invertible, el sistema (2.3) es un sistema lineal compatible determinado, obteniéndose

$$\widehat{\begin{pmatrix} \alpha \\ \beta \end{pmatrix}} = (X^t X)^{-1} X^t \mathbf{y}$$

En efecto,

$$f(\alpha, \beta) = \left(y - X^t \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right)^t \left(y - X^t \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right)$$

Si ahora tomamos $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ como θ:

$$f(\theta) = (y - X\theta)^t (y - X\theta) = \sum_{i=1}^{N} \left(y_i - \mathbf{x}_i^t \theta \right)^2$$

Entonces;

$$\frac{\partial}{\partial \theta_k} f(\theta) = \sum_{i=1}^{N} \left(y_i - \mathbf{x}_i^t \theta \right) (-\mathbf{x}_{ik})$$

Por tanto;

$$\nabla f(\theta) = -2X^t (y - X\theta)$$

$$\nabla f(\theta) = 0 \Leftrightarrow -2X^t (y - X\theta) = 0 \Leftrightarrow X^t (y - X\theta) = 0 \Leftrightarrow X^t y = X^t X\theta \Leftrightarrow \theta = \left(X^t X \right)^{-1} X^t y$$

Capítulo 3

Medidas de bondad de ajuste

Error cuadrático medio:

El error cuadrático medio (MSE o mean squared error en inglés) es una forma de evaluar la diferencia entre un estimador y el valor real de la cantidad que se quiere calcular. El MSE mide el promedio del cuadrado del "error", siendo el error el valor en la que el estimador difiere de la cantidad a ser estimada.
En otras palabras, se construye es estimador muestral de $E((y - \mathbf{X}\beta)^2)$ como

$$MSE = \frac{\sum_{i=1}^{N}(y_i - \hat{y_i})^2}{N}$$

Coeficiente de determinación R^2:

El Coeficiente de Determinación R^2 da la proporción de variación de la variable **y** que es explicada por la variable X (variable predictora o explicativa). Si la proporción es igual a 0, significa que la variable predictora no tiene ninguna capacidad predictiva de la variable a predecir (**y**). Cuanto mayor sea R^2, mejor será la predicción. Si llegara a ser igual a 1 la variable predictora explicaría perfectamente la variación de **y**, y las predicciones no tendrían error.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y_i})^2}{\sum_{i=1}^{N}(y_i - \bar{y_i})^2}$$

Las medidas anteriores indican la bondad del ajuste sobre los propios elementos (x_j,y_j), pero no dan información sobre la bondad del ajuste para una observación (**x**, y) diferente de las de la muestra. En la literatura se han propuesto distintas medidas de bondad de ajuste sobre datos ajenos a la propia muestra, siendo la más usada la validación cruzada.

Validación cruzada con k pliegues:

- Tomar k (por ejemplo k=5, k=10,...)
- Dividir aleatoriamente la muestra en k grupos aproximadamente del mismo tamaño
- Para cada grupo, tomar éste como muestra test, y los restantes como muestra de aprendizaje

- Tomar como error el promedio de los k errores así obtenidos

Validación cruzada con n pliegues (leave one out):

- Se realiza validación cruzada con n pliegues, siendo n el número de regristros de la base de datos
- Tomar como error el promedio de los errores así obtenidos

Capítulo 4

El Lasso

4.1. Formulación como un problema de optimización

El Método Lasso (**L**east **A**bsolute **S**hrinkage and **S**election **O**perator), introducido por Tibshirani (1996) es un método que combina un modelo de regresión con un procedimiento de contracción de algunos parámetros hacia cero y selección de variables, imponiendo una restricción o una penalización sobre los coeficientes de regresión.

Vamos a presentar una formulación del Lasso como problema de optimización basándonos en [4] y [5].

Supongamos que tenemos los datos (x_i,y_i), $i = 1, 2, ..., N$, donde $x_i = (x_{i1}, ..., x_{ip})^t$ son las variables predcitoras y y_i son las respuestas. Sin pérdida de generalidad, podemos considerar que las x_{ij} están estandarizadas, es decir,

$$\sum_i x_{ij}/N = 0,$$

$$\sum_i {x_{ij}}^2/N = 1$$

o dicho de otra forma, tienen media cero y varianza 1. De no verificarse la condición anterior, basta tipificar las variables como parte del preprocesamiento.

Si denotamos $\hat{\beta}=(\hat{\beta_1}, ..., \hat{\beta_p})^t$, la estimación del lasso $(\hat{\alpha}, \hat{\beta})$ se define como la solución óptima del problema de optimización:

$$\begin{array}{ll} \text{mín}_{\alpha,\beta} & \{\sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2\} \\ \text{sujeto a} & \sum_j |\beta_j| \leq t \end{array} \tag{4.1}$$

donde $t \geq 0$ es un parámetro de ajuste.

Fijado β que satisfaga $\sum_j |\beta_j| \leq t$, optimizar en α es un problema de optimización diferenciable en una variable, cuya condición de optimalidad es gradiente igual a cero.

Proposición 4.1.1. *Para todo $t \geq 0$ la solución óptima para α en el problema (4.1) es $\hat{\alpha} = \overline{y}$.*

Demostración

En efecto;

$$\begin{aligned} \tfrac{d}{d\alpha}(\textstyle\sum_{i=1}^{N}(y_i - \alpha - \sum_j \beta_j x_{ij})^2) &= 0 \\ \textstyle\sum_{i=1}^{N} 2(y_i - \alpha - \sum_j \beta_j x_{ij}) &= 0 \\ \textstyle\sum_{i=1}^{N} y_i - N\alpha - \sum_{i,j} \beta_j x_{ij} &= 0 \end{aligned} \tag{4.2}$$

Teniendo en cuenta que por hipótesis:

$$\sum_{i,j} \beta_j x_{ij} = 0$$

entonces,

$$\alpha = \frac{1}{N}\sum_{i=1}^{N} y_i$$

□

Por tanto, podemos asumir sin pérdida de generalidad que $\overline{y} = 0$, y por lo tanto omitir α.

Si recordamos la función de la norma ℓ_q,

$$\|\beta\|_q = (\sum_{i=1}^{p} |\beta_i|^q)^{1/q}$$

podemos reescribir (4.1) como:

$$\begin{aligned} &\text{mín}_{\alpha,\beta} \quad \{\|Y - X\beta\|_2^2\} \\ &\text{sujeto a} \quad \|\beta\|_1 \leq t \end{aligned} \tag{4.3}$$

Obtenemos un modelo equivalente a (4.3) añadiendo una regularización de tipo ℓ_1 en los coeficientes de regresión:

$$\min_{\alpha,\beta}\{\|Y - X\beta\|_2^2 + \lambda\|\beta\|_1\} \tag{4.4}$$

En las expresiones anteriores t y λ son parámetros de regularización o de penalización.

La regularización ℓ_1 agrega una penalización ℓ_1 igual al valor absoluto de la magnitud de los coeficientes. En otras palabras, limita el tamaño de los coeficientes. La norma ℓ_1 puede generar modelos dispersos (es decir, modelos con pocos coeficientes). Algunos coeficientes pueden convertirse en cero y eliminarse.

El cálculo de la solución del problema de optimización (4.1) es un problema de programación cuadrático convexo con restricciones lineales de desigualdad. En la sección 4.3 describiremos algunos algoritmos para la búsqueda de soluciones óptimas.

La siguiente proposición nos prueba que, efectivamente, los problemas (4.3) y (4.4) son equivalentes.

Proposición 4.1.2. *Los problemas (4.3) y (4.4) son equivalentes*

Demostración
Vamos a hallar las condiciones de KKT (Karush-Kuhn-Tucker) [2] de cada uno de los problemas. Para ello vamos a realizar el cambio $\beta = \beta^+ - \beta^-$, β^+, $\beta^- \geq 0$, que corresponde a desdoblar la variable en su parte positiva y su parte negativa.

Consideramos el problema:

$$\min_{\beta} \sum_{i=1}^{N} (y_i - \mathbf{x}_i^t \beta)^2 + \lambda \sum_{i=1}^{p} |\beta_i| \tag{4.5}$$

Haciendo el cambio $\beta = \beta^+ - \beta^-$, nos queda:

$$\begin{array}{ll} \min_{\beta^+,\beta^-} & \sum_{i=1}^{N} (y_i - \mathbf{x}_i^t \beta^+ + \mathbf{x}_i^t \beta^-)^2 + \lambda \sum_{i=1}^{p} \left(\beta_i^+ + \beta_i^-\right) \\ \text{sujeto a} & -\beta^+ \leq 0 \\ & -\beta^- \leq 0 \end{array} \tag{4.6}$$

Definimos la funcion lagrangiana $\mathcal{L}$ como:
$\mathcal{L}(\beta^+, \beta^-, v^+, v^-) = \sum_{i=1}^{N} (y_i \text{ - } {x_i}^t \ \beta^+ + {x_i}^t \ \beta^-)^2 + \lambda \sum_{i=1}^{p} \left(\beta_i^+ + \beta_i^-\right) - v^+\beta^+ - v^-\beta^-$

Entonces las condiciones de KKT de (4.5) son:

$$2\sum_{i=1}^{N} (y_i \text{ - } {x_i}^t \ \beta^+ + {x_i}^t \ \beta^-)(-x_i) + \lambda \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} - v^+ = 0$$

$$2\sum_{i=1}^{N} (y_i \text{ - } {x_i}^t \ \beta^+ + {x_i}^t \ \beta^-)(x_i) + \lambda \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} - v^- = 0$$

$v^+\beta^+ = 0$
$v^-\beta^- = 0$
$v^+,\ v^- \geq 0$

Consideramos ahora el problema:

$$\begin{array}{ll} \min_{\beta} & \sum_{i=1}^{N} (y_i - \mathbf{x}_i^t \beta)^2 \\ \text{sujeto a:} & \sum_{j=1}^{p} |\beta_j| \leq t \end{array} \tag{4.7}$$

Haciendo el cambio $\beta = \beta^+ - \beta^-$, nos queda:

$$\begin{array}{ll} \min_{\beta^+,\beta^-} & \sum_{i=1}^{N} (y_i - \mathbf{x}_i^t \beta^+ + \mathbf{x}_i^t \beta^-)^2 \\ \text{sujeto a:} & \sum_{j=1}^{p} \beta_j^+ + \beta_j^- - t \leq 0 \\ & -\beta^+ \leq 0 \\ & -\beta^- \leq 0 \end{array} \tag{4.8}$$

Su lagrangiano es:
$\mathcal{L}(\beta^+, \beta^-, u, v^+, v^-) = \sum_{i=1}^{N} (y_i \text{ - } {x_i}^t \ \beta^+ + {x_i}^t \ \beta^-)^2 + u(\sum_{j=1}^{p} \beta_j^+ + \beta_j^- - t \ ) - v^+\beta^+ - v^-\beta^-$

Entonces las condiciones de KKT de (4.7) son:

$$2\sum_{i=1}^{N} (y_i \text{ - } {x_i}' \ \beta^+ + {x_i}' \ \beta^-)(-x_i) + \begin{pmatrix} u \\ \vdots \\ u \end{pmatrix} - v^+ = 0$$

$$2\sum_{i=1}^{N} (y_i \text{ - } {x_i}' \ \beta^+ + {x_i}' \ \beta^-)(x_i) + \begin{pmatrix} u \\ \vdots \\ u \end{pmatrix} - v^- = 0$$

$u(\sum_{j=1}^{p} \beta_j^+ + \beta_j^- - t\ ) = 0$
$v^+\beta^+ = 0$
$v^-\beta^- = 0$
$v^+,\ v^+, u \geq 0$

Vamos a demostrar que los dos problemas son equivalentes mostrando que para cada solución de las ecuaciones de KKT (4.5) es posible construir una solución de las ecuaciones KKT para (4.7), y recíprocamente.

Por tanto si tenemos $(\beta_\lambda^+, \beta_\lambda^-, v_\lambda^+, v_\lambda^-)$ solución de KKT (1), tomando

$$t = \sum_{j=1}^{p} \beta_{j\lambda}^+ + \beta_{j\lambda}^-$$

y

$$u = \lambda,$$

$(\beta_\lambda^+, \beta_\lambda^-, v_\lambda^+, v_\lambda^-)$ es también solución de KKT (2).
Recíprocamente, si tenemos $(\beta_t^+, \beta_t^-, v_t^+, v_t^-, u_t)$ solución de KKT (2) para un t fijo, entonces tomando $\lambda = u$, es solución de KKT (1).
Por tanto como las condiciones de KKT son condiciones necesarias y suficientes de optimalidad, podemos admitir que ambos problemas son equivalentes. □

4.2. Predicción y estimación del parámetro t

En esta sección vamos a dar un método para la estimación del parámetro t del Lasso. Podemos encontrar otros métodos de estimación en [4].
Estimamos el error de predicción para el Lasso usando una validación cruzada con k-pliegues, procedimiento descrito en el capítulo 3, por ejemplo con k=10.

Si llamamos

$$s = \frac{t}{\sum_{i=1}^{p} \hat{\beta}_j^o},$$

donde $\hat{\beta}_j^o$ son los estimadores de mínimos cuadrados, y hacemos variar s en una rejilla lo suficientemente fina, entre 0 y 1, para cada valor de s o respecivamente de t, obtenemos mediante validación cruzada un estimador $\hat{e}(t)$ del error cuadrático medio de predicción, como se definió en el capítulo 3. Determinamos así t^*, valor de t con menor $\hat{e}(t)$, y es este el parámetro considerado.

4.3. Algoritmos para encontrar soluciones

Una vez hemos obtenido una estimación de t, a la que denominaremos t^*, procedemos a resolver el problema de optimización;

$$\begin{array}{ll} \text{mín}_\beta & \sum_{i=1}^{N} (y_i - \mathbf{x}_i^t \beta)^2 \\ \text{sujeto a} & \sum_{i=1}^{p} |\beta_i| \leq t^* \end{array} \tag{4.9}$$

Observamos que el problema (4.9) tiene p variables, ya que $\beta \in \mathbb{R}^p$, y una restricción; esta restricción la podemos transformar en 2^p restricciones lineales:

$$\| \beta \|_1 \leq t^*$$

$$\sum_{i=1}^{p} |\beta_i| \leq t^*$$

$$\sum_{i=1}^{p} \beta_i^+ + \beta_i^- \leq t^*$$

$$\sum_{i=1}^{p} u_i \beta_i \leq t^* \qquad \forall (u_1, ..., u_p) \in \{-1, 1\}^p$$

El problema anterior es un problema de optimización cuadrático convexo con 2^p restricciones lineales. Es posible obtener una formulación equivalente con un número lineal en p de restricciones, ampliando el número de variables. Para ello hacemos el cambio:

$$\beta = \beta^+ - \beta^-,$$

teniendo en cuenta que β_i puede expresarse como

$$\beta_i = \beta_i^+ - \beta_i^-,$$

con

$$\beta_i^+, \beta_i^- \geq 0.$$

De donde

$$|\beta_i| = \beta_i^+ + \beta_i^-.$$

Por tanto, (4.9) es equivalente a:

$$\begin{array}{ll} \text{mín}_{\beta^+, \beta^-} & \sum_{i=1}^{N} (y_i - \mathbf{x}_i^t(\beta^+ - \beta^-))^2 \\ \text{sujeto a} & \sum_{i=1}^{p} \left(\beta_i^+ + \beta_i^-\right) \leq t^* \\ & \beta^+, \beta^- \geq 0 \end{array} \tag{4.10}$$

Este problema tiene $2p$ variables ya que $\beta^+, \beta^- \in \mathbb{R}^p$, y $2p + 1$ restricciones.

Capítulo 5

Elastic Net

Aunque el algoritmo LASSO (sección 4) funciona en la mayoría de las ocasiones, tiene algunas limitaciones:

- Cuando $p > N$, es decir, tenemos más variables que observaciones, LASSO selecciona como mucho N variables, debido a la naturaleza de optimización convexa. Esto parece ser una característica limitante para un modelo de selección de variables.
  Además, LASSO no está bien definido a menos que el límite de la norma ℓ_1 de los coeficientes sea menor que un cierto valor.
- Si se tiene un grupo de variables entre las cuales las realaciones a pares son muy altas, entonces LASSO tiende a seleccionar solo una variable de dicho grupo, sin importarle cúal se selecciona.

El objetivo de esta sección es, por tanto, presentar un modelo que funciona tan bien como LASSO y que pueda resolver los problemas mencionados anteriormente.

Se propone una nueva técnica de regularización que llamamos **Elastic Net**, Red Elástica. Veáse [8].

5.1. Definición

De igual manera que en LASSO (sección 4), supongamos que tenemos los datos (x_i,y_i), $i = 1, 2, ..., N$, donde $x_i = (x_{i1}, ..., x_{ip})^t$ son las variables predcitoras y y_i son las respuestas. Admitimos que las x_{ij} están estandarizadas, es decir,

$$\sum_i x_{ij}/N = 0,$$

$$\sum_i {x_{ij}}^2/N = 1$$

o dicho de otra forma, tienen media cero y varianza 1. Además suponemos que la variable respuesta está centrada,

$$\sum_i y_i = 0$$

.

Para cualesquiera λ_1 y λ_2 fijos, no negativos; definimos el criterio de la red elástica de Naïve.

$$\mathcal{L}(\lambda_1, \lambda_2, \beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_2\|\beta\|_2^2 + \lambda_1\|\beta\|_1 \tag{5.1}$$

Donde

$$\mathbf{y} = (y_1, ..., y_N)^t,$$
$$\mathbf{X} = (\mathbf{X}_1|...|\mathbf{X}_p),$$
$$\|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2,$$
$$\|\beta\|_1 = \sum_{j=i}^{p} |\beta_j|.$$

Si denotamos $\hat{\beta}$=$(\hat{\beta}_1, ..., \hat{\beta}_p)^t$, la estimación del Naïve Elastic Net $\hat{\beta}$ se define como la solución óptima del problema de optimización:

$$\text{mín}_\beta \quad \mathcal{L}(\lambda_1, \lambda_2, \beta) \tag{5.2}$$

Este procedimiento puede verse como un método penalizado de mínimo cuadrados. Si denotamos por

$$\alpha = \frac{\lambda_2}{(\lambda_1 + \lambda_2)},$$

entonces hallar $\hat{\beta}$ en la equación (5.1) es equivalente a resolver el problema de optimización:

$$\begin{aligned} &\text{mín}_\beta && \{\|\mathbf{y} - \mathbf{X}\beta\|^2\} \\ &\text{sujeto a} && (1-\alpha)\|\beta\|_1 + \alpha\|\beta\|_2^2 \leq t \end{aligned} \tag{5.3}$$

Llamamos a la función

$$(1-\alpha)\|\beta\|_1 + \alpha\|\beta\|_2^2$$

penalización del Elastic Net, es una combinación convexa de las penalizaciones de LASSO y ridge regression. Cuando $\alpha = 1$, Naïve Elastic Net se convierte simplemente en una ridge regression.

Se va a considerar el caso $\alpha < 1$. Para todo $\alpha \in [0, 1)$, la función penalización de Elastic Net no es derivable en los puntos con alguna coordenada $\beta_j = 0$, y es estrictamente convexa para $\alpha > 0$.

5.2. Solución

Se describe a continuación un método para resolver eficientemente el problema de la red élastica básica. Minimizar la ecuación 5.1, es equivalente a un problema de optimización tipo LASSO.

Lema 5.2.1. *Dados los datos ($\boldsymbol{y}$, $\boldsymbol{X}$) y (λ_1, λ_2), definimos los datos artificiales ($\boldsymbol{y}^*, \boldsymbol{X}^*$) como*

$$\boldsymbol{X}^*_{(n+p)\times p} = (1+\lambda_2)^{-1/2} \begin{pmatrix} \boldsymbol{X} \\ \sqrt{\lambda_2}\boldsymbol{I} \end{pmatrix},$$

$$\boldsymbol{y}^*_{(n+p)} = \begin{pmatrix} \boldsymbol{y} \\ 0 \end{pmatrix}$$

Sea $\gamma = \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}$ *y* $\beta^* = \sqrt{(1+\lambda_2)}\beta$.

Entonces Naïve Elastic Net puede reescribirse como:

$$\mathcal{L}(\gamma, \beta) = \mathcal{L}(\gamma, \beta^*) = \|\boldsymbol{y}^* - \boldsymbol{X}^*\beta^*\|^2 + \gamma\|\beta^*\|_1 \tag{5.4}$$

Si llamamos

$$\hat{\beta^*} = \min_{\beta^*} \mathcal{L}(\gamma, \beta^*)$$

Entonces

$$\hat{\beta} = \frac{1}{\sqrt{(1+\lambda_2)}}\hat{\beta^*}$$

Demostración

Partiendo de la equación 5.4, se quiere llegar a la equación 5.1.

$$\|\mathbf{y}^* - \mathbf{X}^*\beta^*\|^2 + \gamma\|\beta^*\|_1 = \| \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - (1+\lambda_2)^{-1/2} \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I} \end{pmatrix} \sqrt{(1+\lambda_2)}\beta\|^2 + \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}\|\sqrt{1+\lambda_2}\beta\|_1$$

Simplificando nos queda:

$$\|\mathbf{y}^* - \mathbf{X}^*\beta^*\|^2 + \gamma\|\beta^*\|_1 = \| \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I} \end{pmatrix} \beta\|^2 + \lambda_1\|\beta\|_1$$

Operemos con el primer sumando:

$$\| \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I} \end{pmatrix} \beta\|^2 = \left[\begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I} \end{pmatrix} \beta \right]' \left[\begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I} \end{pmatrix} \beta \right] =$$

$$= (y - X\beta)'(y - X\beta) + \left(0 - \sqrt{\lambda_2}I\beta\right)' \left(0 - \sqrt{\lambda_2}I\beta\right) = (y - X\beta)'(y - X\beta) + \left(0 - \sqrt{\lambda_2}\beta' I\right) \left(0 - \sqrt{\lambda_2}I\beta\right) =$$

$$= \|y - X\beta\|^2 + \lambda_2\beta'\beta = \|y - X\beta\|^2 + \lambda_2\|\beta\|^2$$

Por tanto uniendo los dos sumandos obtenidos, tenemos la equación (5.1). □

5.3. Deficiencia del Elastic Net de Naïve

Como método de selección de variables, Elastic Net supera las limitaciones de LASSO expuestas al principio de este capítulo. Sin embargo, la red elástica de Naïve no funciona satisfactoriamente a menos que esté muy cerca de ridge regression o de LASSO. El estimador de la red elástica Naïve es un procedimiento de dos etapas: para cada λ_2 fijo, primero encontramos los coeficientes de ridge regrassion, y luego hacemos la contracción de tipo LASSO. La contracción doble no ayuda a reducir mucho las varianzas. A continuación se va a mejorar el rendimiento predictivo de la red elástica al corregir esta doble contracción.

5.4. Estimación de Elastic Net

Dados los datos ($\mathbf{y}$, $\mathbf{X}$) y (λ_1, λ_2),y los datos artificiales ($\mathbf{y}^*, \mathbf{X}^*$) definidos en el lema 5.2.1, Naïve Elastic Net resuelve el problema de tipo LASSO:

$$\hat{\beta^*} = \min_{\beta^*} \|\mathbf{y}^* - \mathbf{X}^*\beta^*\|^2 + \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}\|\beta^*\|_1 \tag{5.5}$$

Las estimaciones de Elastic Net (corregidas) $\hat{\beta}$ se definen como:

$$\hat{\beta}(\text{elastic net}) = \sqrt{1+\lambda_2}\hat{\beta^*}$$

Recordemos que $\hat{\beta}(\text{naïve elastic net}) = \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}\hat{\beta^*}$, así

$$\hat{\beta}(\text{elastic net}) = (1+\lambda_2)\hat{\beta}(\text{naïve elastic net})$$

Por tanto, el coeficiente de elastic net es simplemente una reescala del coeficiente de naïve elastic net.

Dicha transformación de escala conserva la propiedad de selección variable de la red elástica denaïve y es la forma más simple de deshacer la contracción.

Por lo tanto, todas las buenas propiedades de naïve elastic net se conservan para elastic net.

Se va a presentar ahora un teorema que da otra representación de elastic net.

Teorema 5.4.1. *Dados los datos ($\boldsymbol{y}$, $\boldsymbol{X}$) y (λ_1, λ_2), entonces el estimador de elastic net $\hat{\beta}$ viene dado por:*

$$\hat{\beta}(\textit{elastic net}) = \min_{\beta} \beta^t \left(\frac{\boldsymbol{X}^t\boldsymbol{X} + \lambda_2\boldsymbol{I}}{1+\lambda_2}\right)\beta - 2\boldsymbol{y}^t\boldsymbol{X}\beta + \lambda_1\|\beta\|_1 \tag{5.6}$$

Demostración
Sea $\hat{\beta}$ el estimador de elastic net. Por las definiciones dadas en el lema 5.2.1, y la equación 5.5, tenemos:

$$\hat{\beta} = \min_{\beta} |\mathbf{y}^* - \mathbf{X}^*\frac{\beta}{\sqrt{1+\lambda_2}}|^2 + \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}|\frac{\beta}{\sqrt{1+\lambda_2}}|_1 = \min_{\beta}\left(\frac{\mathbf{X}^{*t}\mathbf{X}^*}{1+\lambda_2}\right)\beta - 2\frac{\mathbf{y}^{*t}\mathbf{X}^*}{\sqrt{1+\lambda_2}} + \mathbf{y}^{*t}\mathbf{y}^* + \frac{\lambda_1|\beta|_1}{1+\lambda_2} \tag{5.7}$$

Sustituyendo las identidades:

$$\mathbf{X}^{*t}\mathbf{X}^* = \left(\frac{\mathbf{X}^t\mathbf{X} + \lambda_2}{1+\lambda_2}\right)$$

$$\mathbf{y}^{*t}\mathbf{X}^* = \frac{\mathbf{y}^t\mathbf{X}}{\sqrt{1+\lambda_2}}$$

$$\mathbf{y}^{*t}\mathbf{y}^* = \mathbf{y}^t\mathbf{y}$$

en la equación (5.7), se tiene

$$\hat{\beta} = \min_{\beta} \frac{1}{1+\lambda_2} \left\{ \beta^t \left(\frac{\mathbf{X}^t\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2} \right) \beta - 2\mathbf{y}^t\mathbf{X}\beta + \lambda_1 \|\beta\|_1 \right\} + \mathbf{y}^t\mathbf{y} =$$

$$= \min_{\beta} \beta^t \left(\frac{\mathbf{X}^t\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2} \right) \beta - 2\mathbf{y}^t\mathbf{X}\beta + \lambda_1 \|\beta\|_1$$

Se puede ver que,

$$\hat{\beta}(\text{lasso}) = \min_{\beta} \beta^t(\mathbf{X}^t\mathbf{X})\beta - 2\mathbf{y}^t\mathbf{X}\beta + \lambda_1 \|\beta\|_1 \tag{5.8}$$

□

El teorema 5.4.1 interpreta el elastic net como una estabilización de LASSO. Nótese que $\hat{\Sigma} = \mathbf{X}^t\mathbf{X}$ es una versión encogida de la matriz de correlaciones Σ, y

$$\frac{\mathbf{X}^t\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2} = (1-\gamma)\hat{\Sigma} + \gamma\mathbf{I}$$

con $\gamma = \frac{\lambda_2}{1+\lambda_2}$ que contrae $\hat{\Sigma}$ hacia la matriz identidad.

Juntas, las equaciones (5.6) y (5.8), muestran que reescalar después de la penalización de elastic net es matemáticamente equivalente a reemplazar Σ por su versión encogida en LASSO.

Capítulo 6

El Lars

6.1. El Algortimo Lars

El algoritmo Least angle regression (LAR) está fuertemente conectado con el Lasso, y de hecho proporciona un procedimiento muy eficiente para computar el Lasso. Ver [6].

El procedimiento de selección de variables denominado forward stepwise regression construye un modelo secuencialmente, agregando una variable en cada vez. En cada paso, identifica la mejor variable para incluir en el conjunto activo, y luego actualiza el ajuste de mínimos cuadrados incluyendo todas las variables activas.

Least angle regression usa una estrategia similar a la empleada por el procedimiento anterior, pero incorporando los predictores de forma gradual, de modo que cada predictor toma parte en el modelo, tanto como se "merezca". En el primer paso, identifica la variable más correlacionada con la respuesta. En lugar de ajustar esta variable por completo, LAR mueve el coeficiente de esta variable continuamente hacia su valor de mínimos cuadrados (provocando que su correlación con el residuo disminuya en valor absoluto). Tan pronto como otra variable la "alcanza" en términos de correlación con el residuo, el proceso se detiene. La segunda variable luego se une al conjunto activo, y sus coeficientes se mueven juntos de manera que sus correlaciones evolucionan de modo coincidente, disminuyendo su valor. Este proceso continúa hasta que todas las variables están en el modelo, y termina con el ajuste de mínimos cuadrados completo.

Algoritmo LAR:

1. Tipificar los predictores de forma que tengan media cero y norma unidad. Comenzar con el residuo $r = y - \overline{y}$, y los coeficientes $\beta_1 = \beta_2 = ... = \beta_p = 0$
2. Encontrar el predictor x_j más correlacionado con r.
3. Mover β_j desde 0 hacia su coeficiente de mínimos cuadrados, hasta que otro predictor x_k tenga tanta correlación con el residuo actual como x_j.
4. Mover β_j y β_k en la dirección definida por su coeficiente de mínimos cuadrados conjunto del residuo actual, hasta que otro predictor x_l tenga la misma correlación con el residuo actual.
5. Continuar el proceso hasta que los p predictores hayan sido añadidos.

Supongamos que A_k es el conjunto activo de variables al inicio de la itereación k, y llamemos β_{A_k} al vector de coeficientes para estas variables en esta iteración. Entonces habrá k-1 valores no nulos, y el recién añadido será cero. Si

$$r_k = y - \mathbf{X}_{A_k}\beta_{A_k}$$

es el residuo actual, entonces puede probarse que la dirección del paso 4 del algoritmo para esa iteración es

$$\delta_k = \left(\mathbf{X}_{A_k}^t \mathbf{X}_{A_k}\right)^{-1} \mathbf{X}_{A_k}^t r_k$$

El coeficiente pasa a ser

$$\beta_{A_k}(\alpha) = \beta_{A_k} + \alpha\delta_k$$

Si el vector de ajuste al comienzo de esta iteración es $\hat{\mathbf{f}}_k$, entonces este es

$$\hat{\mathbf{f}}_k(\alpha) = \hat{\mathbf{f}}_k + \alpha\mathbf{u}_k,$$

donde

$$\mathbf{u}_k = \mathbf{X}_{A_K}\delta_k$$

es la nueva dirección.
El nombre "least angle" ("ángulo mínimo") surge de una interpretación geométrica de este proceso; $\mathbf{u}_k$ hace el ángulo más pequeño (e igual) con cada uno de los predictores en A_k.

Una sencilla modificación del algoritmo LAR permite obtener la trayectoria completa de soluciones de Lasso:

Modificación del Algoritmo LAR (para el Lasso):
En el paso **4**, añadimos un apartado **a)** Si un coeficiente distinto de cero llega a cero, borrar su variable del conjunto activo de variables y volver a calcular la dirección actual de mínimos cuadrados conjuntos.

El algoritmo LAR es extremadamente eficiente, ya que requiere, en esencia, el mismo esfuerzo computacional que un sólo ajuste de mínimos cuadrados usando los p predictores. Least angle regression siempre requiere p iteraciones para llegar a los mínimos cuadrados completos estimados. El Lasso puede tener más de p iteraciones; aún así ambos son muy similares. El algoritmo LAR con la modificación indicada es una manera muy eficiente de computar el Lasso.

Ahora damos un argumento heurístico de por qué estos procedimientos son tan similares.

Aunque el algoritmo LAR se establece en términos de correlaciones, si las características de entrada están estandarizadas, es equivalente y más fácil trabajar con productos escalares.
Supongamos que A es el conjunto activo de variables en alguna itereación del algoritmo; entonces podemos expresar la correlación como el producto escalar

$$x_j^t(y - \mathbf{X}\beta) = \gamma \cdot s_j \quad \forall j \in A \tag{6.1}$$

donde $s_j \in \{-1, 1\}$ indican el signo del producto, y γ es el valor común (absoluto).
Por tanto

$$|x_j^t(y - \mathbf{X}\beta)| \leq \gamma \quad \forall j \notin A$$

Ahora consideramos el Lasso

$$R(\beta) = \frac{1}{2}\| y - \mathbf{X}\beta \|_2^2 + \lambda\| \beta \|_1$$

Sea B el conjunto activo de variables en la solución para un valor dado de λ. Para estas variables $R(\beta)$ es diferenciable,

$$x_j^t(y - \mathbf{X}\beta) = \lambda \cdot sign(\beta_j) \forall j \in B \tag{6.2}$$

Comparando las ecuaciones (6.1) y (6.2), observamos que sólo son idénticas en el caso de que el signo de β_j coincida con el del producto. Esto es por lo que el algoritmo LAR y el Lasso comienzan a diferenciarse cuando un coeficiente activo pasa por cero; en tal caso la condición (6.2) no se cumple para esta variable y es expulsada del conjunto activo B.

6.2. El paquete Lars en R

A continuación vamos a dar una breve descripción del paquete LARS (veáse [1]) existente en el software estadístico R, veáse [3]:

---

cv.lars

Calcula la curva de error de una validación cruzada de k pliegues para lars

---

Descripción
Calcula el error cuadrático medio de predicción de una validación cruzada con k pliegues, para lars, lasso, y forward stagewise. **Uso**
$cv.lars(x, y, K = 10, index, trace = FALSE, plot.it = TRUE, se = TRUE, type = c(\text{“}lasso''\text{, “}lar''\text{, “}forward.stagewise''\text{, “}stepwise''), mode = c(\text{“}fraction''\text{, “}step''), ...)$
Argumentos

x	Introducir en lars
y	Introducir en lars
k	Número de pliegues
index	Valores de abscisas en los que se debe calcular la curva CV. El modo puede ser “fraction” o “step”.
trace	Mostrar cálculos.
plot.it	Dibujarlo.
se	Incluir bandas de error estándar.
type	Tipo de ajuse de lars, con lasso predeterminado
mode	Esto se refiere a los datos que se usan para la validación cruzada. El predeterminado es “fraction” para type=”lasso”o type=”forward.stagewise”. Para type=”lar” o type=”stepwise” el predeterminado es “step”.
...	Argumentos adicionales para lars

---

lars
Realiza el ajuste de los modelos de LARS, Lasso y Forward Stagewise

---

Descripción
Implementa todas las variantes de Lasso, y proporcionan toda la secuencia de coeficientes y ajustes, a partir de cero, hasta el ajuste de mínimos cuadrados.

Uso

$lars(x, y, type = c(\text{“}lasso''\text{, “}lar''\text{, “}forward.stagewise''\text{, “}stepwise''), trace = FALSE, normalize = TRUE, intercept = TRUE, Gram, eps = .Machine\$double.eps, max.steps, use.Gram = TRUE)$

Argumentos

x	Matriz de predictores
y	Respuesta
type	“lasso”, “lar”, “forward.stagewise” o “stepwise”, el valor predeterminado es “lasso”
trace	Si es TRUE, lars imprime su progreso
normalize	Si es TRUE, cada variable se estandariza para tener la norma de la unidad L2. El valor predeterminado es TRUE.
intercept	Si es TRUE, se incluye una intersección en el modelo (y no se penaliza); de lo contrario, no se intersepta. El valor predeterminado es TRUE.
Gram	La matriz X^tX; útil para ejecuciones repetidas.
eps	Un cero efectivo.
max.steps	Limíta el número de pasos del algoritmo; el valor predeterminado es $8\,\text{mín}(m, n - intercept)$, con m el número de variables, y n el número de muestras. Para type = “lar” o type = “stepwise”, el número máximo de pasos es $\text{mín}(m, n - intercept)$. Para type = “lasso” y especialmente type = “forward.stagewise”, puede haber muchos más términos, porque aunque no más de $\text{mín}(m, n - intercept)$ variables pueden estar activas en cada paso, las variables son eliminadas y añadidas frecuentemente en el algoritmo.
use.Gram	Cuando el número m de variables es muy grande, es decir, más grande que N, entonces es posible que no desee que LARS precompute la matriz de Gram. El valor predeterminado es use.Gram = TRUE.

---

plot.lars
Método de dibujo para objetos de lars

Descripción
Realiza una representación gráfica de un ajuste de lars. Por defecto representa el recorrido completo de los coeficientes.

Uso
$plot(x, xvar = c(\text{“}norm''\text{, “}df''\text{, “}arc.length''\text{, “}step''), breaks = TRUE, plottype = c(\text{“}coefficients''\text{, “}Cp''), omit.zer$
$TRUE, eps = 1e-10, ...)$

Argumentos

---

predict.lars
Hacer predicciones o extraer coeficientes de un modelo de lars ajustado

---

Descripción
Mientras lars() proporciona la trayectoria de soluciones completa, predict.lars permite extraer una predicción en un punto particular a lo largo de la trayectoria.

Uso
$predict(object, newx, s, type = c(\text{“}fit''\text{, “}coefficients''), mode = c(\text{“}step''\text{, “}fraction''\text{, “}norm''\text{, “}lambda''), ...)$

x	Objeto lars
xvar	El tipo de variable x frente a lo que dibujamos. xvar=norm (valor por defecto) frente a la norma $l1$ del vector de coeficientes. como una fracción de la norma máxima $l1$. xvar=step representa frente al número de pasos (que es esencialmente grados de libertad de LAR, no para Lasso o Forward Stagewise). xvar=arc.length representa frente al arco de longitud del vector ajustado. xvar=df representa frente al estimado df, que es el tamaño del conjunto activo en cada paso.
breaks	Si es TRUE, se trazan líneas verticales en cada punto de corte con los trayectos de los coeficientes.
plottype	Cualquiera de los coeficientes (por defecto) o Cp. El gráfico e coeficientes muestra la ruta de cada coeficiente en función de la fracción de norma o Df. El gráfico Cp muestra la curva Cp.
omit.zeros	Cuando el número de variables es mucho mayor que el número de observaciones, muchos coeficientes nunca serán distintos de cero; esta lógica (el valor por defecto es TRUE) evita trazar estos coeficientes cero.
eps	Cero efectivo, el valor predeterminado es 1e-10.
...	Argumentos adicionales para la representación gráfica. Se puede usar para configurar xlims, cambiar colores, anchos de línea, etc.

Argumentos

object	Objeto lars ajustado
newx	Si type="fit", entonces newx deben ser los valores de x en los que se requiere el ajuste. Si type="coefficients" entonces newx puede ser omitido.
s	Un valor, o un vector de valores, por defecto (mode="step"), s toma valores entre 0 y p.
type	Si type="fit", devuelve los valores ajustados. Si type="coefficients", devuelve los coeficientes del modelo.
mode	Puede ser mode="step", "fraction", "norm", "lambda".
...	Argumentos adicionales para predict.lars

---

summary.lars
Método de resumen para objetos lars

Descripción
Produce un resumen ANOVA para objetos lars.

Uso
$summary(object, sigma2 = NULL, ...)$

Argumentos

object	Objeto lars
sigma2	medida de varianza opcional (para $p > n$).
...	Argumentos adicionales.

Además encontramos el siguiente paquete de datos;

---

diabetes
Sangre y otras medidas en diabéticos

Descripción
El dataframe de diabetes tiene 442 filas y 3 columnas.
Formato
Este paquete de datos contiene las siguientes estructuras de datos (o variables):

- **x** una matriz con 10 columnas
- **y** un vector numérico
- **x2** una matriz con 64 columnas

Detalles
La matrix **x** ha sido estandarizada para tener norma ℓ_2 unidad en cada columna y media cero. La matriz **x2** consiste en la matriz **x** y ciertas iteraciones.

El proceso a seguir sería el siguiente:

- Aplicamos lars() al conjunto de datos dados, la matriz de datos x y el vector y. Podemos usar cualquiera de los modelos LARS, Lasso o Forward Stagewise
- Con predict.lars() podemos predecir nuevos valores.
- Usando plot.lars() podemos dar una representación de cómo va evolucionando el modelo aplicado (LARS, Lasso o Forward Stagewise) en cada iteración.
- Además con summary.lars podemos obtener un resumen ANOVA.

Capítulo 7

Ejemplos numéricos

7.1. Primer ejemplo numérico con la base de datos Prostata

7.1.1. Descripción del paquete de datos

Nombre: Prostate data, [7].
En este conjunto de datos tenemos 97 casos, y 9 variables, 8 variables de entrada y 1 de salida.

Variables de entrada:

- lcavol
- lweight
- age
- lbph
- svi
- lcp
- gleason
- pgg45

Variables de salida:

- lpsa

7.1.2. Ajuste por mínimos cuadrados (OLS)

Con la base de datos descrita anteriormente en la sección (7.1.1), realizamos un ajuste por mínimos cuadrados (OLS), usando el programa R [3].

Comenzamos leyendo los datos con las órdenes:

prostata= read.csv(”*prostate.txt*”, $sep =$ ””, $dec =$ ”.”, $header = TRUE$)
prostata=prostata[,2:10]

```
Call:
lm(formula = lpsa ~ ., data = prostata)

Residuals:
     Min       1Q   Median       3Q      Max
-1.64151 -0.29859  0.00946  0.35860  1.72489

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.335361   1.290786   0.260  0.79562
lcavol       0.561927   0.085745   6.553 3.82e-09 ***
lweight      0.618958   0.196110   3.156  0.00220 **
age         -0.022709   0.010838  -2.095  0.03905 *
lbph         0.093191   0.056552   1.648  0.10299
svi          0.721279   0.236070   3.055  0.00298 **
lcp         -0.145262   0.089347  -1.626  0.10761
gleason      0.032612   0.151806   0.215  0.83041
pgg45        0.006542   0.004355   1.502  0.13669
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6828 on 87 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.6568,Adjusted R-squared:  0.6252
F-statistic: 20.81 on 8 and 87 DF,  p-value: < 2.2e-167
```

Figura 7.1: Summary(ols)

A continución ajustamos por mínimos cuadrados (OLS):
ols$< -lm(lpsa\ .,data = prostata)$

Mostramos los coeficientes obtenidos y alguna información adicional sobre el ajuste proporcionado por el comando ***summary*** en la Figura 7.1

Observaciones:
La columna correspondiente a $Pr(>|t|)$, en la Figura 7.1 muestra los p-valores resultantes del contraste de hipótesis:

$$\begin{cases} H_0 : \beta_i = 0 \\ H_1 : \beta_i \neq 0 \end{cases} \tag{7.1}$$

Por tanto, podemos sacar como conclusiones que las variables más influyentes en el modelo son, por orden respectivo a la significancia de sus p-valores, **lcavol**, **lweight**, **svi** y **age**. Por el contrario, las variables restantes, **lbph**, **lcp**, **gleason** y **pqq45**, según nuestro contraste de hipótesis bajo el modelo lineal de mínimo cuadrados (OLS) no resultan influyentes en el modelo considerado.

Por otra parte, se obtene un coeficiente R^2 de 0.6, por lo que puede considerarse que nuestro conjunto de datos se encuentra moderadamente bien aproximado por el modelo lineal.

Ahora mostramos un conjunto de 4 gráficas, Figuras 7.2 a 7.5, resumiendo el modelo OLS mediante la función ***plot***.

En la Figura 7.2 se muestran los valores ajustados frente a los residuos; se observa que prácticamente la línea roja se ajusta a la línea horizontal $y = 0$, lo cual nos indica que tenemos un buen ajuste lineal. En cuanto a las observaciones atípicas, vemos que destacan tres obsevaciones: la **69**, la **95**, y la **39**.

En la Figura 7.3 se ha realizado una prueba de normalidad que ha resultado satisfactorio, ya que nuestra gráfica casi que se ajusta a la línea discontinua, separándose un poco más en los extremos debido a las observaciones atípicas, que se corresponden con las mismas observadas en la gráfica anterior, como era de esperar.

La Figura 7.4 representa una prueba sobre homocedasticidad; como podemos contemplar en la gráfica la línea se ajusta a una línea horizontal, lo cual nos lleva a admitir la homocedasticidad.

Por último la Figura 7.5 nos sirve para detectar los valores influyentes, (**69**, **95** y **47**), los cuales no tendrían por qué coincidir con los valores atípicos observados en las Figuras 7.2 y 7.3. Sin embargo vemos que la observación **69** además de ser atípica es influyente. Si repitiéramos el estudio sin estas observaciones influyentes puede que obtuviéramos un mejor ajuste.

Con la orden ***confint(ols)***, en la Figura 7.6, mostramos intervalos de confianza para los coeficientes β_i de las 8 variables de nuestro modelo. Se puede observar que aquellos que contienen el 0 son los correspondientes a las variables que no resultaron significativas debido a los p-valores del contraste de hipótesis de la Figura 7.1.

Ahora realizamos predicciones, primero mostrando el intervalo de confianza con la orden:
$pred1 = predict.lm(ols, interval = \text{“}confidence'')$
Y luego mostrando el intervalo de predicción con la orden:
$pred2 = predict.lm(ols, interval = \text{“}predicition'')$

Mostramos la cabecera de los resultados obtenidos en las Figuras 7.7 y 7.8 respectivamente.

En la Figura 7.9 se presenta una gráfica donde mostramos las predicciones para cada uno de los 97 datos (en azul) junto con sus correspondientes extremos de los intervalos de confianza (en rojo) y de predicción (en negro). Se observa en dicha Figura que los intervalos de predicción son mayores que los de confianza.

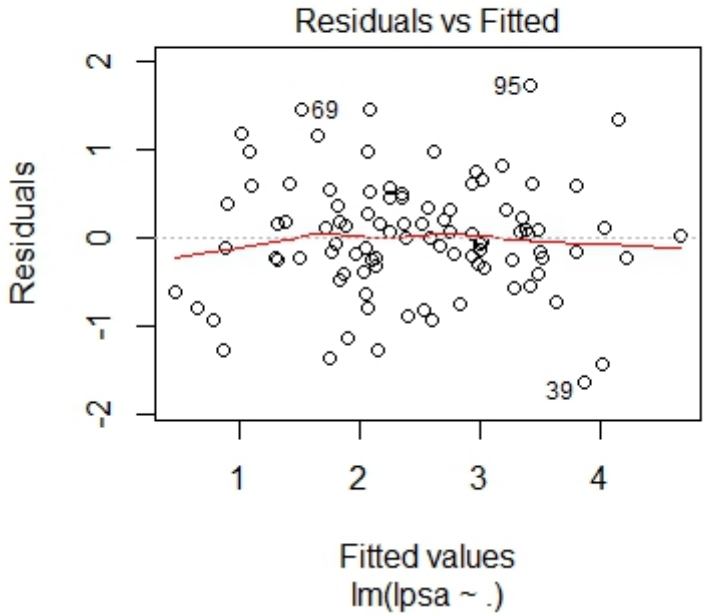

Figura 7.2: ¿Cómo de bien se ajusta al modelo lineal?

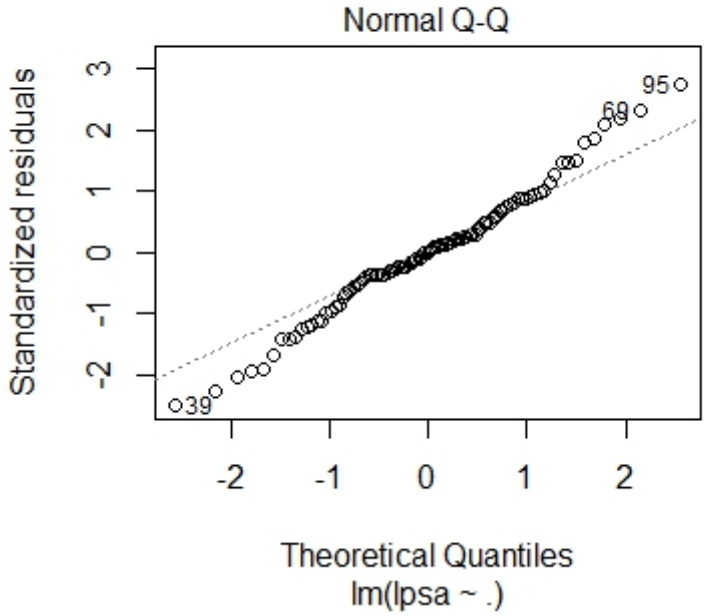

Figura 7.3: Test de normalidad

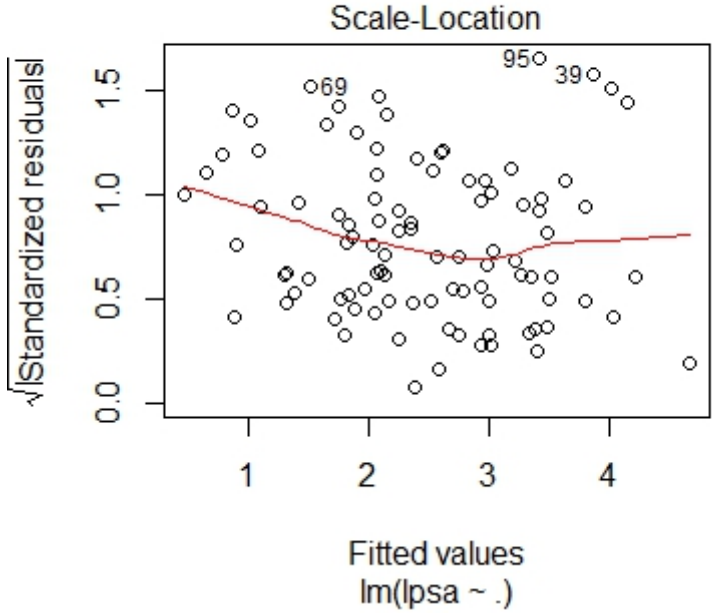

Figura 7.4: Test de homocedasticidad

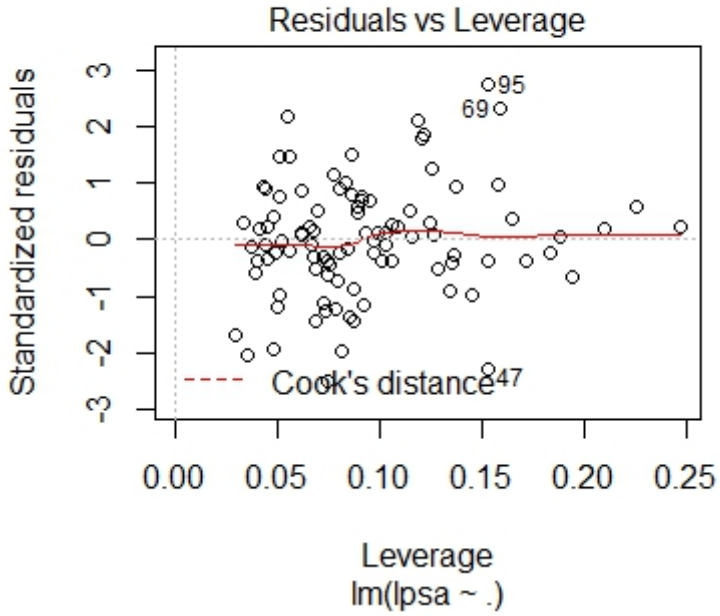

Figura 7.5: Valores influyentes

```
                  2.5 %       97.5 %
(Intercept) -2.230215599  2.900937931
lcavol       0.391500007  0.732354924
lweight      0.229168589  1.008747400
age         -0.044251425 -0.001167433
lbph        -0.019212662  0.205593674
svi          0.252063625  1.190494007
lcp         -0.322848191  0.032324649
gleason     -0.269118777  0.334342226
pgg45       -0.002114623  0.015199240
```

Figura 7.6: Intervalos de confianza OLS

```
        fit         lwr       upr
1 0.8229078  0.42719338 1.2186222
2 0.7612550  0.39842393 1.1240860
3 0.4416131 -0.08839695 0.9716232
4 0.6199877  0.23164067 1.0083347
5 1.7315458  1.47163266 1.9914590
6 0.8434007  0.44083654 1.2459649
```

Figura 7.7: Predicciones OLS con intervalo de confianza

```
        fit        lwr      upr
1 0.8229078 -0.6224273 2.268243
2 0.7612550 -0.6754252 2.197935
3 0.4416131 -1.0461080 1.929334
4 0.6199877 -0.8233477 2.063323
5 1.7315458  0.3173469 3.145745
6 0.8434007 -0.6038248 2.290626
```

Figura 7.8: Predicciones OLS con intervalo de predicción

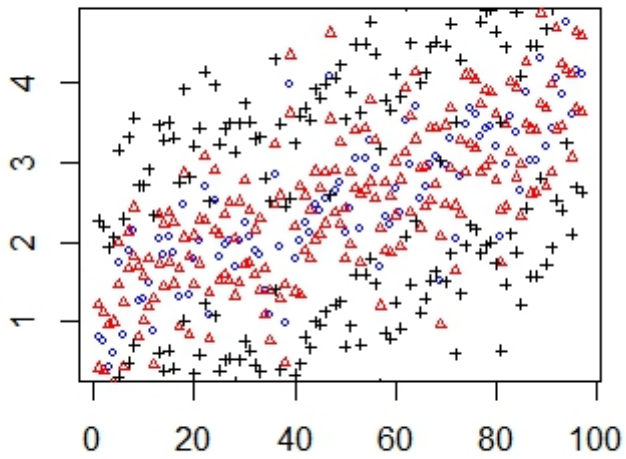

Figura 7.9: Predicciones con sus correspondientes intervalos de confianza y predicción

7.1.3. Ajuste mediante LASSO

Vamos a realizar ahora un ajuste empleando la metodología LASSO:

Para ello comenzamos cargando la librería lars, y los datos:

library(lars)

X = as.matrix(prostata[,1:8])
X

y = prostata[,9]
y

Con la siguiente orden realizamos el ajuste por LASSO

lasso <− lars(X,y,type=c("lasso"))

plot(lasso)

En la Figura 7.10 vemos la gráfica obtenida mediante un plot del ajuste con LASSO, en ella vamos observando, de derecha a izquierda, las variables cuyos coeficientes se van haciendo cero

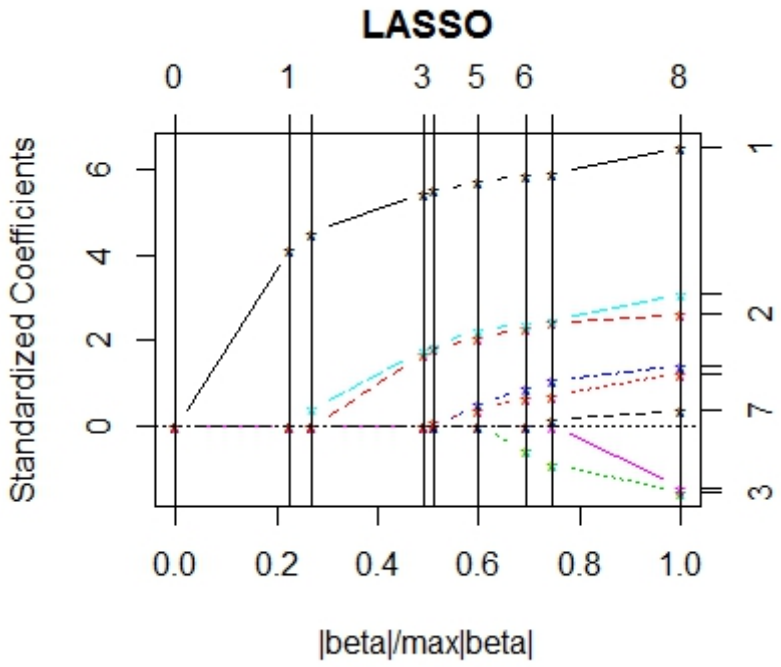

Figura 7.10: Gráfica obtenida mediante el ajuste por LASSO

en cada paso.

Seguidamente vamos a realizar predicciones con la siguiente orden:
predicciones $< -predict.lars(lasso, X)$

En la Figura 7.11 se muestran predicciones obtenidas mediante el ajuste por LASSO, mostrándose en la columna [,1] predicciones con una sola variable en el modelo, y de manera equivalente con las demás columnas, hasta llegar a la columna [,8] con predicciones con las 8 variables en el modelo. Tenemos una novena columna en la que se cuenta el término independiente.

Se pueden ver los coeficientes que da lasso en la Figura 7.12, se observa como en cada paso un coeficiente distinto se hace cero, la primera variable cuyo coeficiente se hace cero es **lcp**, vemos que no coincide con la variable menos significativa obtenida con el ajuste OLS, en cuyo caso esta variable correspondía a **gleason**.

Nuestro objetivo ahora es, una vez extraídas las predicciones para cada valor de λ, mostrar en un conjunto de gráficas la comparativa entre la predicción obtenida mediante OLS y las correspondientes a cada valor de λ obtenidas con LASSO.

En la Figura 7.13 observamos que la línea de puntos negra obtenida con el último valor de λ, para el que ya todos los coeficientes se han hecho cero, se corresponde, como ya sabíamos, con el ajuste OLS. Otra observacion interesante puede hacerse sobre la línea de puntos color

```
\$fit
        [,1]   [,2]   [,3]   [,4]   [,5]    [,6]    [,7]    [,8]    [,9]
 [1,] 2.4784 1.7888 1.7014 1.1253 1.0749 0.89649 0.87166 0.85537 0.8229078
 [2,] 2.4784 1.6408 1.5391 1.1499 1.1161 0.96135 0.89613 0.85893 0.7612550
 [3,] 2.4784 1.8135 1.7285 1.1266 1.0814 0.91998 0.71696 0.61021 0.4416131
 [4,] 2.4784 1.5658 1.4570 1.0359 1.0000 0.83958 0.76985 0.73047 0.6199877
 [5,] 2.4784 2.2645 2.2228 2.0208 1.9982 1.87988 1.80983 1.76638 1.7315458
 [6,] 2.4784 1.6209 1.5174 1.0872 1.0501 0.88935 0.88298 0.87903 0.8434007
 [7,] 2.4784 2.2594 2.2172 2.0305 2.0092 1.96358 1.93036 1.90673 1.9001676
 [8,] 2.4784 2.2437 2.2000 2.0362 2.0169 2.00667 2.04809 2.06638 2.1330020
 [9,] 2.4784 1.7186 1.6244 1.3411 1.3155 1.17685 1.21434 1.23522 1.2546269
[10,] 2.4784 2.0757 2.0159 1.6956 1.6645 1.52654 1.43207 1.37557 1.2953383
```

Figura 7.11: Predicciones con LASSO

```
        lcavol   lweight          age       lbph        svi
 [1,] 0.0000000 0.0000000  0.000000000 0.00000000 0.00000000
 [2,] 0.3573072 0.0000000  0.000000000 0.00000000 0.00000000
 [3,] 0.3916323 0.0000000  0.000000000 0.00000000 0.09772183
 [4,] 0.4729686 0.4010287  0.000000000 0.00000000 0.44189300
 [5,] 0.4772307 0.4343405  0.000000000 0.00000000 0.46205106
 [6,] 0.4945025 0.4904080  0.000000000 0.03525646 0.55370843
 [7,] 0.5067509 0.5431376 -0.008040524 0.06070074 0.58841287
 [8,] 0.5115481 0.5748987 -0.012590999 0.07452697 0.61037529
 [9,] 0.5643413 0.6220198 -0.021248185 0.09671252 0.76167340
            lcp    gleason        pgg45
 [1,]  0.0000000 0.00000000 0.0000000000
 [2,]  0.0000000 0.00000000 0.0000000000
 [3,]  0.0000000 0.00000000 0.0000000000
 [4,]  0.0000000 0.00000000 0.0000000000
 [5,]  0.0000000 0.00000000 0.0003752489
 [6,]  0.0000000 0.00000000 0.0013866469
 [7,]  0.0000000 0.00000000 0.0022898666
 [8,]  0.0000000 0.01704893 0.0024820450
 [9,] -0.1060509 0.04922793 0.0044575118
```

Figura 7.12: Coeficientes obtenidos por le método LASSO

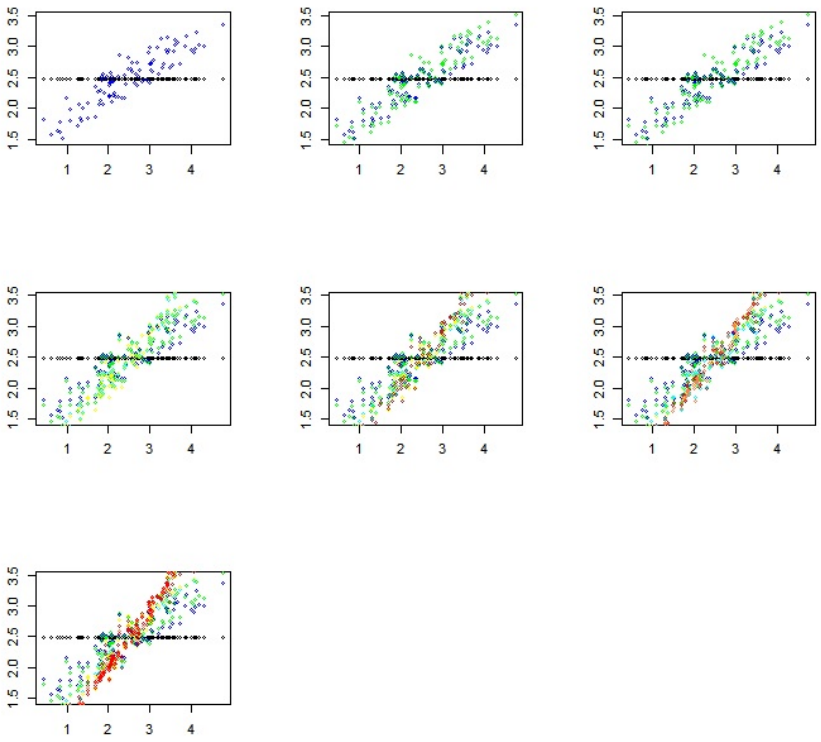

Figura 7.13: Gráficas predicciones con OLS frente a predicciones con LASSO para cada valor de λ

rojo, que forma una diagonal, esta se corresponde con el modelo lineal, ya que sólo tenemos el término independiente en este caso y, por tanto, no tenemos penalización ninguna en el método LASSO.

Seguidamente vamos a mostrar en la Tabla 7.1 los distintos valores de λ obtenidos, las proporciones correspondientes con respecto al valor de λ_1 y los errores cuadráticos medios según las predicciones obtenidas con LASSO para cada valor de λ $(\lambda_1, ..., \lambda_8)$ respectivamente.

Valores de λ_i	Proporciones $\frac{\lambda_i}{\lambda_1}$	ECM
$\lambda_1 = 8.3067969$	1	1.1483635
$\lambda_2 = 4.1805708$	0.503271	0.8874379
$\lambda_3 = 3.5705887$	0.429839	0.8509970
$\lambda_4 = 1.4068330$	0.169359	0.7197101
$\lambda_5 = 1.2293599$	0.147994	0.7126039
$\lambda_6 = 0.6286376$	0.075677	0.6909422
$\lambda_7 = 0.3630874$	0.04371	0.6782377
$\lambda_8 = 0.2164061$	0.026052	0.6739077

Cuadro 7.1: Tabla λ_i, proporciones λ_i/λ_1 y ECM.

7.1.4. Stepwise Regresion

Definición

El método estadístico **Stepwise Regresion** se puede realizar de varias maneras, en concreto tres; la primera de ellas consiste en ir probando variable a variable (independientes) si es significativa, en caso afirmativo se incluye en el modelo. La segunda se basa en incluir todas las variables en el modelo e ir elimando aquellas que no sean significativas. Por tercer y último lugar se pude hacer una combinación de ambos métodos.

El conjunto de pruebas estadísticas que determinan si una variable es o no significativa es muy amplio, se puede emplear alguno de los siguientes:

- Pruebas con el estadístico F de Fisher
- Pruebas con el estadístico t de Student
- R cuadrado ajustado

El objetivo es encontrar un conjunto de variables independientes que influyan significativamente en la variable dependiente.

Implementado en R

Para finalizar este primer ejemplo numérico se va a realizar un ajuste con el procedimiento Stepwise, para comparar los resultados con los anteriormente obtenidos con OLS o LASSO.

```
Call:
lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi, data = prostata)

Residuals:
     Min       1Q   Median       3Q      Max
-1.86717 -0.37725  0.01257  0.40252  1.44544

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.49473    0.87652   0.564  0.57385
lcavol       0.54400    0.07463   7.289 1.11e-10 ***
lweight      0.58821    0.19790   2.972  0.00378 **
age         -0.01644    0.01068  -1.540  0.12692
lbph         0.10122    0.05759   1.758  0.08215 .
svi          0.71490    0.20653   3.461  0.00082 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6988 on 91 degrees of freedom
Multiple R-squared:  0.6526,Adjusted R-squared:  0.6335
F-statistic: 34.19 on 5 and 91 DF,  p-value: < 2.2e-16
```

Figura 7.14: Coeficientes obtenidos mediante el ajuste Step

En la Figura 7.14 se muestran los coeficientes obtenidos mediante el método Step. Vemos que las variables más significativas en el modelo según Step son: **lcavol**, **svi** y **lweight**, mientras que las variables más significativas con OLS eran **lcavol**, **lweight**, **svi** y **age**.

En cuanto a la variable menos significativa, con Step es la variable **age**, que no coincide con la obtenida con OLS, en cuyo caso esta variable correspondía a **gleason**, y tampoco coincide con la primera variable cuyo coeficiente se hace cero en LASSO, que es **lcp**. Vemos que no todas las variables se encuentran en el modelo, ya que el método usado, Stepwise, ha eliminado las que a priori no ha considerado significativas.

7.1.5. Elastic Net

Se va a hacer uso el paquete ***data(glmnet)***.

Con la siguiente orden se presenta en la Figura 7.15 la gráfica obtenida mediante elastic net para $\alpha = 1$, es decir en el caso en el que coincide con LASSO.

plot(glmnet(X,y, alpha = 1))

Por tanto esta gráfica debería coincidir con la obtenida mediante el método LASSO, Figura 7.10, vemos que aunque son similares no son exactamente iguales; esto es debido a que en la gráfica obtenida con LASSO los coeficientes estan estandarizados, además en el eje **y** se ha

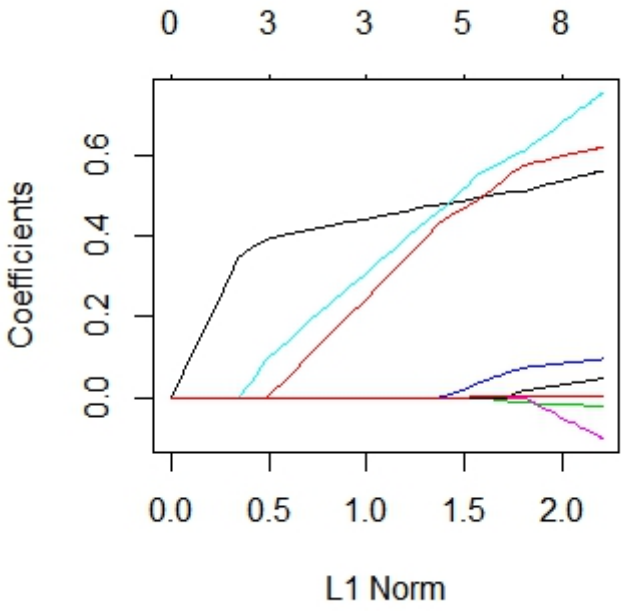

Figura 7.15: Elastic Net $\alpha = 1$

divido por la norma del máximo β, sin embargo en la obtenida con ***glmnet*** el eje **y** representa la norma ℓ_1.

Se han realizado las operaciones necesarias para que los ejes **x** e **y** representen lo mismo en ambas gráficas, de esta forma podemos observar como la Figura 7.15 y la Figura 7.16 coinciden, es decir se obtiene el mismo resultado aplicando un ajuste del tipo LASSO que uno del tipo Elastic Net con el parámetro $\alpha = 1$.

A continuación se muestra en la Figura 7.17 la gráfica obtenida con ***glmnet*** para $\alpha = 0$, es decir en el caso en el que coincide con ridge regression.

Basándonos en una particion de valores de α entre 0 y 1, se ha representado para cada uno de estos valores de α los distintos valores de λ frente al error cuadrático medio para cada uno. En la Figura 7.18 se observan estas gráficas para todos los valores de α considerados en la partición, de manera que el color **negro** se corresponde con $\alpha = 0$ y el color **verde limón** se corresponde al valor de $\alpha = 1$.

7.2. Segundo ejemplo numérico con la base de datos mtcars

7.2.1. Descripción del paquete de datos

Nombre: Motor Trend Car Road Tests.
En este conjunto de datos tenemos 32 casos, y 11 variables, 10 variables de entrada y 1 de salida.

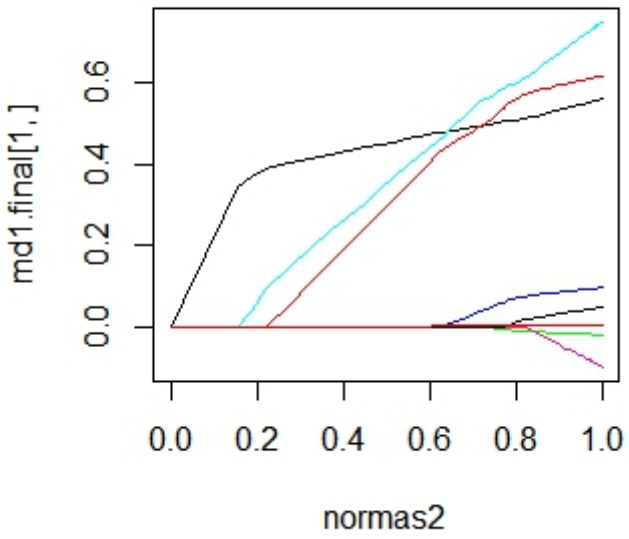

Figura 7.16: Gráfica del ajuste por LASSO una vez hechos los ajustes en los ejes

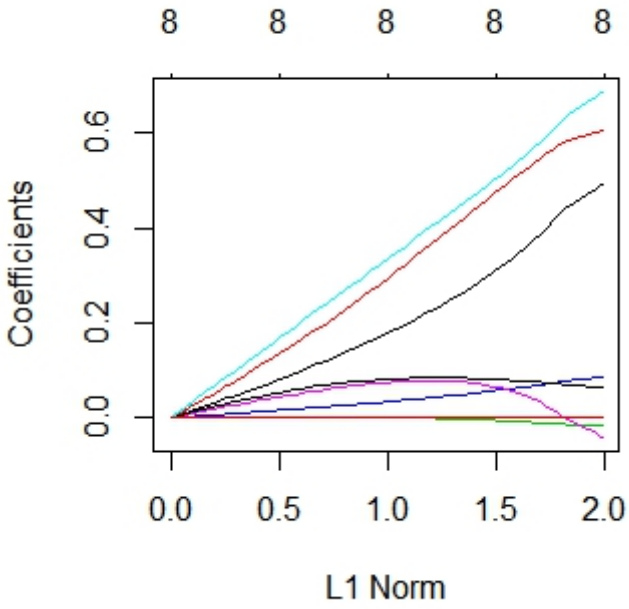

Figura 7.17: Elastic Net para $\alpha = 0$

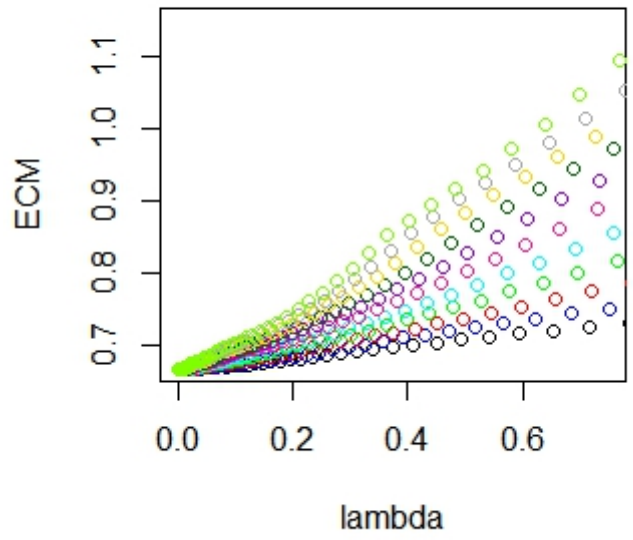

Figura 7.18: ECM frente a λ para cada valor de la partición de α

Los datos fueron extraídos de la revista Motor Trend US de 1974, y comprenden el consumo de combustible y 10 aspectos del diseño y rendimiento del automóvil para 32 automóviles (modelos 1973-74).

Variables de entrada:

- cyl: Número de cilindros
- disp: Desplazamiento (cu.in.)
- hp: Potencia bruta
- drat: Relación del eje trasero
- wt: Peso (1000 lbs)
- qsec: Tiempo de 1/4 milla
- vs: V/S
- am: Transmisión (0 = automático, 1 = manual)
- gear: Número de marchas adelante
- carb: Cantidad de carburadores

Variables de salida:

- mpg: Miles / (US) galón

```
Call:
lm(formula = mpg ~ ., data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4506 -1.6044 -0.1196  1.2193  4.6271

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 12.30337   18.71788   0.657   0.5181
cyl         -0.11144    1.04502  -0.107   0.9161
disp         0.01334    0.01786   0.747   0.4635
hp          -0.02148    0.02177  -0.987   0.3350
drat         0.78711    1.63537   0.481   0.6353
wt          -3.71530    1.89441  -1.961   0.0633 .
qsec         0.82104    0.73084   1.123   0.2739
vs           0.31776    2.10451   0.151   0.8814
am           2.52023    2.05665   1.225   0.2340
gear         0.65541    1.49326   0.439   0.6652
carb        -0.19942    0.82875  -0.241   0.8122
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.65 on 21 degrees of freedom
Multiple R-squared:  0.869,Adjusted R-squared:  0.8066
F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```

Figura 7.19: Summary(OLS)

7.2.2. Ajuste por mínimos cuadrados (OLS)

Con la base de datos descrita anteriormente en la sección (7.2.1), realizamos un ajuste por mínimos cuadrados (OLS), usando el programa R [3].

Comenzamos leyendo los datos con las órdenes:

data(mtcars)
mtcars

A continución ajustamos por mínimos cuadrados (OLS):
OLS$< -lm(mpg\ ., data = mtcars)$

Mostramos los coeficientes obtenidos y alguna información adicional sobre el ajuste proporcionado por el comando ***summary*** en la Figura 7.19

Observaciones:
La columna correspondiente a $Pr(> |t|)$, en la Figura 7.19 muestra los p-valores resultantes del

contraste de hipótesis correspondiente a la equación (7.1).

Debido al bajo número de observaciones, en concreto 32, los resultados de dicho contraste de hipótesis no pueden asegurar que ninguna variable sea significativa.

Por otra parte, se obtene un coeficiente R^2 de 0.8, por lo que puede considerarse que nuestro conjunto de datos se encuentra muy bien aproximado por el modelo lineal.

Ahora mostramos un conjunto de 4 gráficas, Figuras 7.20 a 7.23, resumiendo el modelo OLS mediante la función ***plot***.

En la Figura 7.20 se muestran los valores ajustados frente a los residuos; se observa que la línea roja no se ajusta a la línea horizontal $y = 0$, lo cual nos indica que no tenemos un buen ajuste lineal. En cuanto a las observaciones atípicas, vemos que destacan tres observaciones: la **Chyrster Imperial**, la **Fiat 128**, y la **Toyota Corolla**.

En la Figura 7.21 se ha realizado una prueba de normalidad que ha resultado satisfactorio, ya que nuestra gráfica casi que se ajusta a la línea discontinua, separándose un poco más en los extremos debido a las observaciones atípicas, **Chyrster Imperial**, **Fiat 128**, y **Ford Pantera L**.

La Figura 7.22 representa una prueba sobre homocedasticidad; como podemos contemplar en la gráfica la línea conforme va avanzando en el eje x va dejando de ajustarse a una línea horizontal, lo cual nos lleva decir que podríamos admitir, o no, la homocedasticidad.

Por último la Figura 7.23 nos sirve para determinar los valores influyentes, (**Chyrster Imperial**, **Merc 230**, y **Ford Pantera L**), los cuales no tendrían por qué coincidir con los valores atípicos observados en las Figuras 7.20 y 7.21. Sin embargo vemos que la observaciones **Chyrter Imperial** y **Ford Pantera L** además de ser atípicas son influyentes. Si repitiéramos el estudio sin estas observaciones influyentes puede que obtuviéramos un mejor ajuste.

Con la orden ***confint(ols)***, en la Figura 7.24, mostramos intervalos de confianza para los coeficientes β_i de las 10 variables de nuestro modelo. Se puede observar que todos ellos contienen el 0, ya que ninguna variable resultó significativa con respecto a los p-valores del contraste de hipótesis de la Figura 7.19.

Ahora realizamos predicciones, primero mostrando el intervalo de confianza con la orden:
$pred1 = predict.lm(OLS, interval = \text{“}confidence'')$
Y luego mostrando el intervalo de predicción con la orden:
$pred2 = predict.lm(OLS, interval = \text{“}predicition'')$

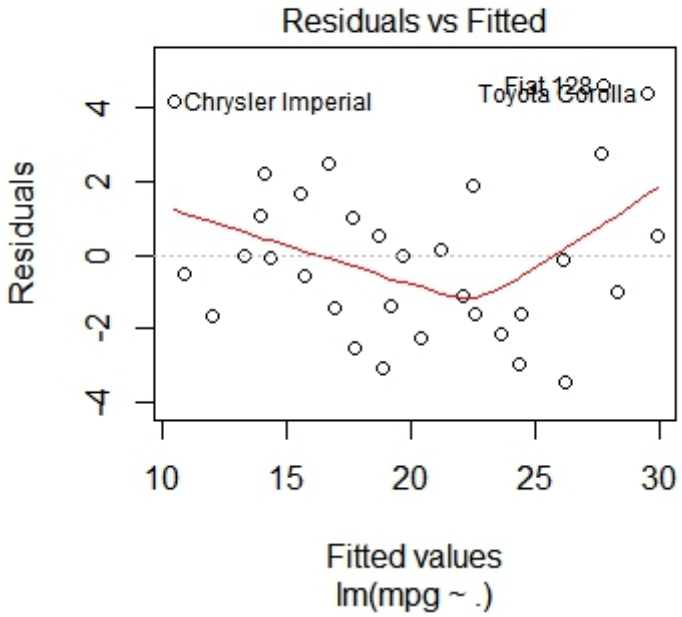

Figura 7.20: ¿Cómo de bien se ajusta al modelo lineal?

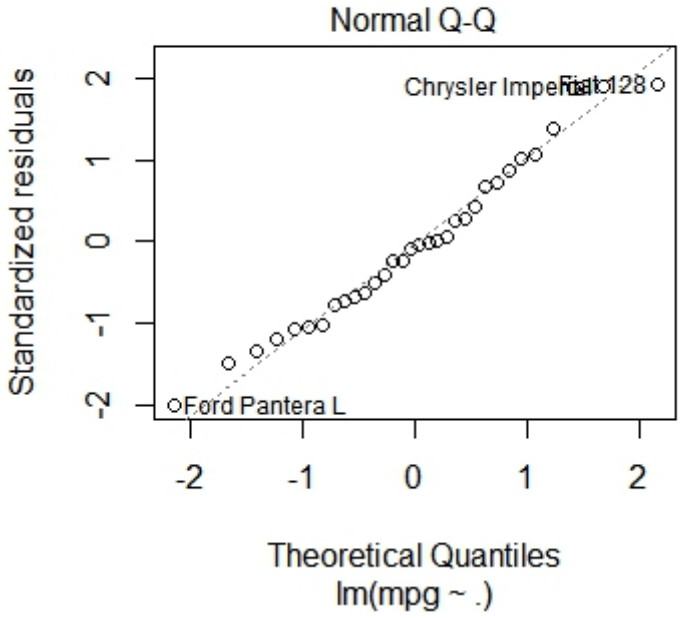

Figura 7.21: Test de normalidad

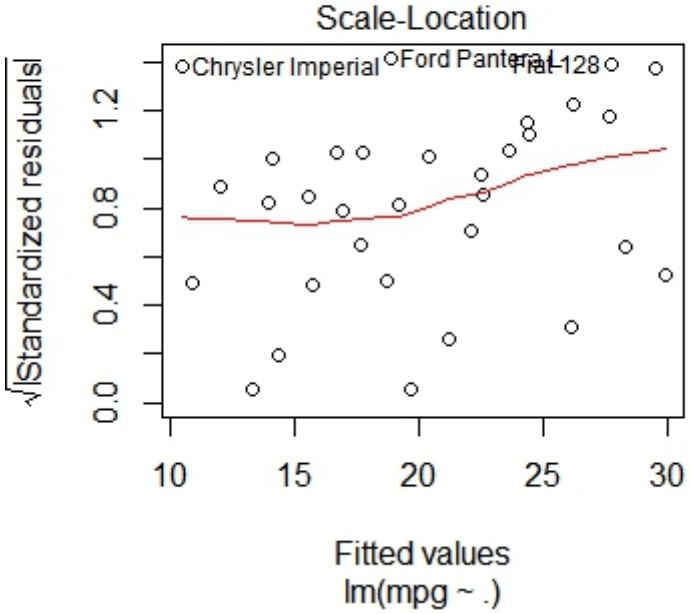

Figura 7.22: Test de homocedasticidad

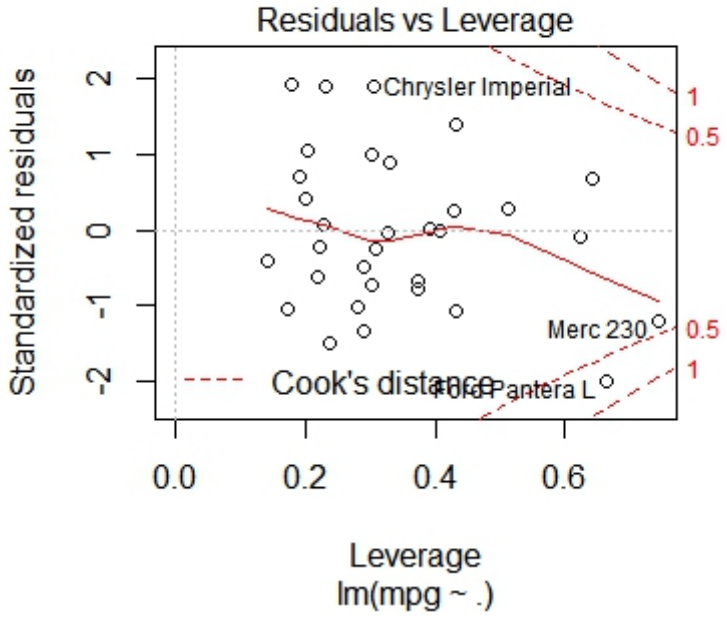

Figura 7.23: Valores influyentes

```
                   2.5 %      97.5 %
(Intercept) -26.62259745 51.22934576
cyl          -2.28468553  2.06180457
disp         -0.02380146  0.05047194
hp           -0.06675236  0.02378812
drat         -2.61383350  4.18805545
wt           -7.65495413  0.22434628
qsec         -0.69883421  2.34091571
vs           -4.05880242  4.69432805
am           -1.75681208  6.79726585
gear         -2.44999107  3.76081711
carb         -1.92290442  1.52406591
```

Figura 7.24: Intervalos de confianza OLS

```
                      fit      lwr      upr
Mazda RX4         22.59951 19.56821 25.63080
Mazda RX4 Wag     22.11189 19.14278 25.08099
Datsun 710        26.25064 23.55729 28.94400
Hornet 4 Drive    21.23740 18.60726 23.86755
Hornet Sportabout 17.69343 15.23168 20.15519
Valiant           20.38304 17.45482 23.31126
```

Figura 7.25: Predicciones OLS con intervalo de confianza

Mostramos la cabecera de los resultados obtenidos en las Figuras 7.25 y 7.26 respectivamente.

En la Figura 7.27 se presenta una gráfica donde mostramos las predicciones para cada uno de los 32 datos (en azul) junto con sus correspondientes extremos de los intervalos de confianza (en rojo) y de predicción (en negro). Se observa en dicha Figura que los intervalos de predicción son mayores que los de confianza.

7.2.3. Ajuste mediante LASSO

Vamos a realizar ahora un ajuste empleando la metodología LASSO:

```
                      fit      lwr      upr
Mazda RX4         22.59951 16.30950 28.88951
Mazda RX4 Wag     22.11189 15.85162 28.37215
Datsun 710        26.25064 20.11635 32.38494
Hornet 4 Drive    21.23740 15.13060 27.34421
Hornet Sportabout 17.69343 11.65724 23.72963
Valiant           20.38304 14.14206 26.62402
```

Figura 7.26: Predicciones OLS con intervalo de predicción

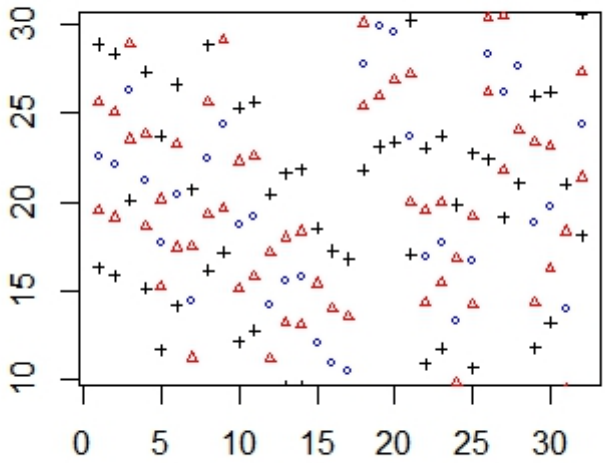

Figura 7.27: Predicciones con sus correspondientes intervalos de confianza y predicción

Para ello comenzamos cargando la librería lars, y los datos:

library(lars)

X = as.matrix(mtcars[,2:11])
X

y = mtcars[,1]
y

Con la siguiente orden realizamos el ajuste por LASSO

lasso <− lars(X,y,type=c("lasso"))

plot(lasso)

En la Figura 7.28 vemos la gráfica obtenida mediante un plot del ajuste con LASSO; en ella vamos observando, de derecha a izquierda, las variables cuyos coeficientes se van haciendo cero en cada paso.

Seguidamente vamos a realizar predicciones con la siguiente orden:
predicciones $< -predict.lars(lasso, X)$

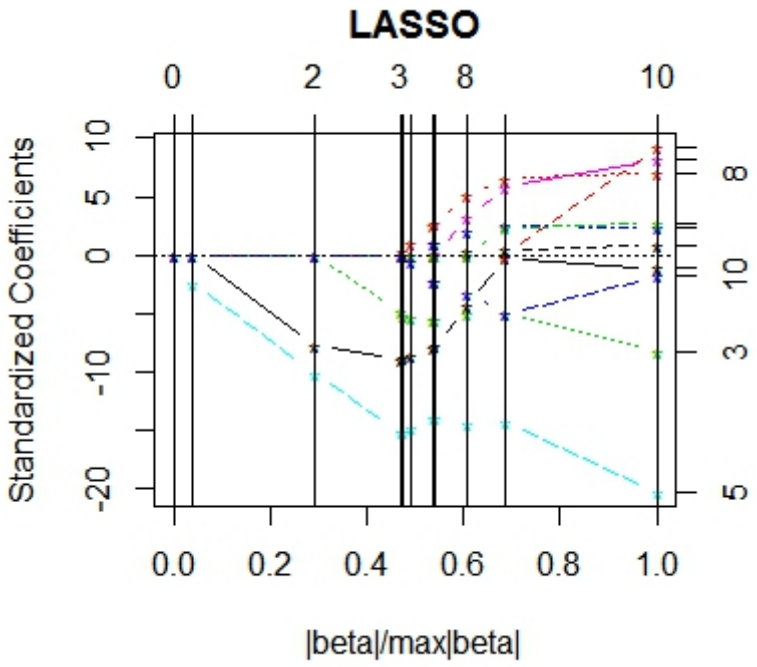

Figura 7.28: Gráfica obtenida mediante el ajuste por LASSO

En la Figura 7.29 se muestran predicciones obtenidas mediante el ajuste por LASSO, mostrándose en la columna [,1] predicciones con una sola variable en el modelo, y de manera equivalente con las demás columnas, hasta llegar a la columna [,10] con predicciones con las 10 variables en el modelo. Tenemos una undécima columna en la que se cuenta el término independiente.

Se pueden ver los coeficientes que da lasso en la Figura 7.30. Se observa como en cada paso un coeficiente distinto se hace cero, siendo **disp** la primera variable que se hace cero. Vemos que no coincide con la variable menos significativa obtenida con el ajuste OLS, en cuyo caso esta variable correspondía a **cyl**.

Nuestro objetivo ahora es, una vez extraidas las predicciones para cada valor de λ, mostrar en un conjunto de gráficas la comparativa entre la predicción obtenida mediante OLS y las correspondientes a cada valor de λ obtenidas con LASSO.

En la Figura 7.31 observamos que la línea de puntos negra obtenida con el último valor de λ, para el que ya todos los coeficientes se han hecho cero, se corresponde, como ya sabíamos, con el ajuste OLS. Otra observacion interesante puede hacerse sobre la línea de puntos color gris, que forma una diagonal; esta se corresponde con el modelo lineal, ya que sólo tenemos el término independiente en este caso y, por tanto, no tenemos penalización ninguna en el método LASSO.

```
\$fit
                      [,1]     [,2]     [,3]     [,4]     [,5]
Mazda RX4          20.09062 20.35275 21.34704 22.38634 22.47359
Mazda RX4 Wag      20.09062 20.24083 20.87291 21.67700 21.76752
Datsun 710         20.09062 20.48441 23.46131 25.22413 25.30105
Hornet 4 Drive     20.09062 20.09161 20.24073 20.73122 20.71805
Hornet Sportabout  20.09062 19.99286 18.26590 17.49132 17.44463
Valiant            20.09062 19.98409 19.78519 20.11331 20.10776
Duster 360         20.09062 19.93581 18.02419 16.23903 16.13144
Merc 240D          20.09062 20.10258 21.84368 23.19846 23.20620
Merc 230           20.09062 20.12014 21.91805 22.88985 22.86758
Merc 280           20.09062 19.99286 19.82237 19.93992 19.91802

                      [,6]     [,7]     [,8]     [,9]    [,10]
Mazda RX4          22.52190 22.62626 22.62699 22.62498 22.43174
Mazda RX4 Wag      21.82859 21.97761 21.98579 22.12817 22.08966
Datsun 710         25.50871 25.98864 26.00462 26.33751 26.53051
Hornet 4 Drive     20.73993 20.71085 20.71221 20.77016 20.81104
Hornet Sportabout  17.41509 17.35258 17.34940 17.27600 17.26399
Valiant            20.14319 20.05847 20.06511 20.22306 20.41130
Duster 360         15.96617 15.51290 15.48714 14.96746 14.25542
Merc 240D          23.16100 23.02228 23.00016 22.55885 22.42223
Merc 230           22.81183 22.71607 22.74662 23.30503 24.00179
Merc 280           19.76163 19.45627 19.43807 19.12292 19.01016

                     [,11]
Mazda RX4          22.59951
Mazda RX4 Wag      22.11189
Datsun 710         26.25064
Hornet 4 Drive     21.23740
Hornet Sportabout  17.69343
Valiant            20.38304
Duster 360         14.38626
Merc 240D          22.49601
Merc 230           24.41909
Merc 280           18.69903
```

Figura 7.29: Predicciones con LASSO

```
              cyl       disp          hp      drat         wt
 [1,]  0.00000000 0.00000000  0.00000000 0.0000000  0.0000000
 [2,]  0.00000000 0.00000000  0.00000000 0.0000000 -0.4388814
 [3,] -0.77823487 0.00000000  0.00000000 0.0000000 -1.8593484
 [4,] -0.89348176 0.00000000 -0.01272378 0.0000000 -2.7817250
 [5,] -0.88264027 0.00000000 -0.01361750 0.0000000 -2.7689159
 [6,] -0.87453982 0.00000000 -0.01387639 0.0000000 -2.7188832
 [7,] -0.80330129 0.00000000 -0.01458639 0.3190785 -2.5437212
 [8,] -0.78662353 0.00000000 -0.01450121 0.3355466 -2.5492262
 [9,] -0.43758100 0.00000000 -0.01303707 0.6612247 -2.6438976
[10,] -0.02346152 0.00000000 -0.01296937 0.8686652 -2.6307453
[11,] -0.11144048 0.01333524 -0.02148212 0.7871110 -3.7153039
            qsec         vs        am     gear        carb
 [1,] 0.00000000 0.00000000 0.0000000 0.000000  0.00000000
 [2,] 0.00000000 0.00000000 0.0000000 0.000000  0.00000000
 [3,] 0.00000000 0.00000000 0.0000000 0.000000  0.00000000
 [4,] 0.00000000 0.00000000 0.0000000 0.000000  0.00000000
 [5,] 0.00000000 0.00000000 0.1080349 0.000000  0.00000000
 [6,] 0.00000000 0.00000000 0.3503975 0.000000 -0.06205513
 [7,] 0.00000000 0.00000000 0.9008953 0.000000 -0.25354960
 [8,] 0.01581314 0.00000000 0.9458436 0.000000 -0.25862663
 [9,] 0.31674621 0.09740664 1.8580262 0.000000 -0.35907255
[10,] 0.58707798 0.14623478 2.3395925 0.576363 -0.55904252
[11,] 0.82104075 0.31776281 2.5202269 0.655413 -0.19941925
```

Figura 7.30: Coeficientes obtenidos por le método LASSO

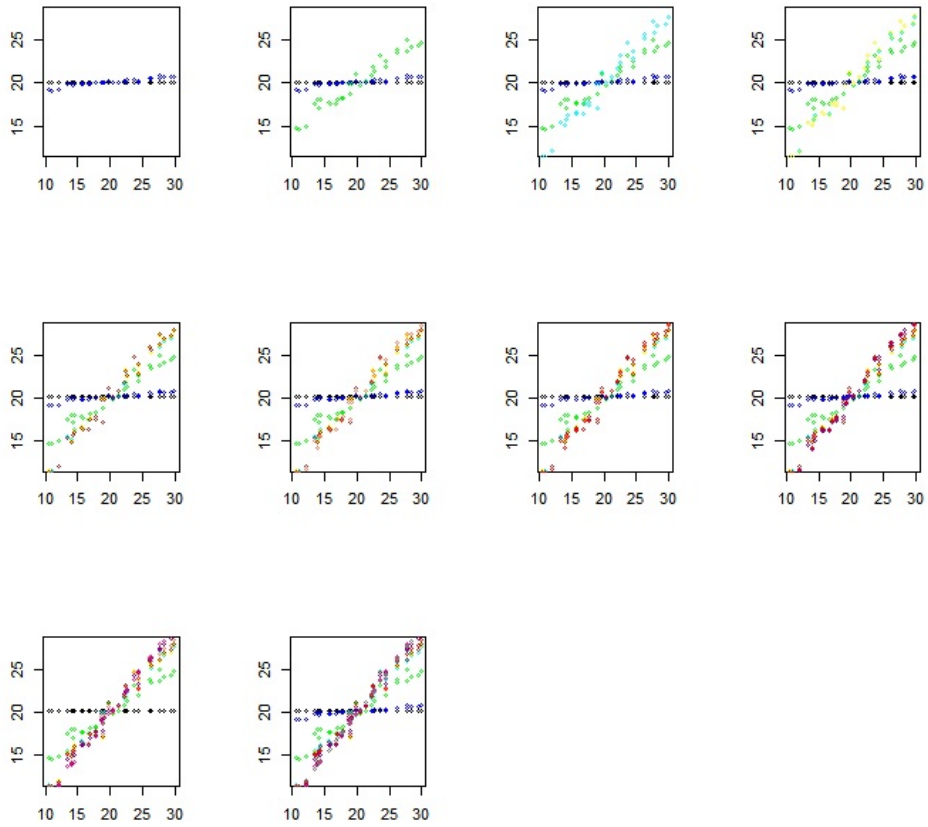

Figura 7.31: Gráficas predicciones con OLS frente a predicciones con LASSO para cada valor de λ

Seguidamente vamos a mostrar en la Tabla 7.2 los distintos valores de λ obtenidos, las proporciones correspondientes con respecto al valor de λ_1 y los errores cuadráticos medios según las predicciones obtenidas con LASSO para cada valor de λ ($\lambda_1, ..., \lambda_1 0$) respectivamente.

Valores de λ_i	Proporciones $\frac{\lambda_i}{\lambda_1}$	ECM
$\lambda_1 = 29.1157217$	1	5.932030
$\lambda_2 = 26.7247746$	0.917881	5.569267
$\lambda_3 = 12.9310216$	0.444125	3.440507
$\lambda_4 = 3.8096949$	0.130847	2.463283
$\lambda_5 = 3.5312331$	0.121283	2.441593
$\lambda_6 = 3.0968059$	0.106362	2.399790
$\lambda_7 = 1.9545598$	0.067131	2.305524
$\lambda_8 = 1.9073386$	0.065509	2.300060
$\lambda_9 = 0.9688393$	0.033275	2.217276
$\lambda_1 0 = 0.2172834$	0.007463	2.177358

Cuadro 7.2: Tabla λ_i, proporciones λ_i/λ_1 y ECM.

7.2.4. Stepwise Regresion

Para finalizar este segundo ejemplo numérico se va a realizar un ajuste con el procedimiento Stepwise, para comparar los resultados con los anteriormente obtenidos con OLS o LASSO.

En la Figura 7.32 se muestran los coeficientes obtenidos mediante el método Step. Vemos que las variables más significativas en el modelo según Step son: **lcavol**, **svi** y **lweight**, mientras que las variables más significativas con OLS eran **lcavol**, **lweight**, **svi** y **age**.

En cuanto a la variable menos significativa, con Step es la variable **am**, que realmente se considera significativa según el contraste de hipótesis, pero es la menos significativa, no coincide con la obtenida con OLS, en cuyo caso esta variable correspondía a **cyl**, y tampoco coincide con la primera variable cuyo coeficiente se hace cero en LASSO, que es **disp**. Vemos que no todas las variables se encuentran en el modelo, ya que el método usado, Stepwise, ha eliminado las que a priori no ha considerado significativas.

7.2.5. Elastic Net

Se va a hacer uso el paquete ***data(glmnet)***.

Con la siguiente orden se presenta en la Figura 7.33 la gráfica obtenida mediante elastic net para $\alpha = 1$, es decir en el caso en el que coincide con LASSO.

```
Call:
lm(formula = mpg ~ wt + qsec + am, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4811 -1.5555 -0.7257  1.4110  4.6610

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.6178     6.9596   1.382 0.177915
wt           -3.9165     0.7112  -5.507 6.95e-06 ***
qsec          1.2259     0.2887   4.247 0.000216 ***
am            2.9358     1.4109   2.081 0.046716 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.459 on 28 degrees of freedom
Multiple R-squared:  0.8497,Adjusted R-squared:  0.8336
F-statistic: 52.75 on 3 and 28 DF,  p-value: 1.21e-11
```

Figura 7.32: Coeficientes obtenidos mediante el ajuste Step

plot(glmnet(X,y, alpha = 1))

Por tanto esta gráfica debería coincidir con la obtenida mediante el método LASSO, Figura 7.28, vemos que aunque son similares no son exactamente iguales; esto es debido a que en la gráfica obtenida con LASSO los coeficientes estan estandarizados, además en el eje **y** se ha divido por la norma del máximo β, sin embargo en la obtenida con ***glmnet*** el eje **y** representa la norma ℓ_1.

Se han realizado las operaciones necesarias para que los ejes **x** e **y** representen lo mismo en ambas gráficas, de esta forma podemos observar como la Figura 7.33 y la Figura 7.34 coinciden, es decir se obtiene el mismo resultado aplicando un ajuste del tipo LASSO que uno del tipo Elastic Net con el parámetro $\alpha = 1$.

A continuación se muestra en la Figura 7.35 la gráfica obtenida con ***glmnet*** para $\alpha = 0$, es decir en el caso en el que coincide con ridge regression.

Basándonos en una particion de valores de α entre 0 y 1, se ha representado para cada uno de estos valores de α los distintos valores de λ frente al error cuadrático medio para cada uno. En la Figura 7.36 se observan estas gráficas para todos los valores de α considerados en la partición, de manera que el color **negro** se corresponde con $\alpha = 0$ y el color **verde limón** se corresponde al valor de $\alpha = 1$.

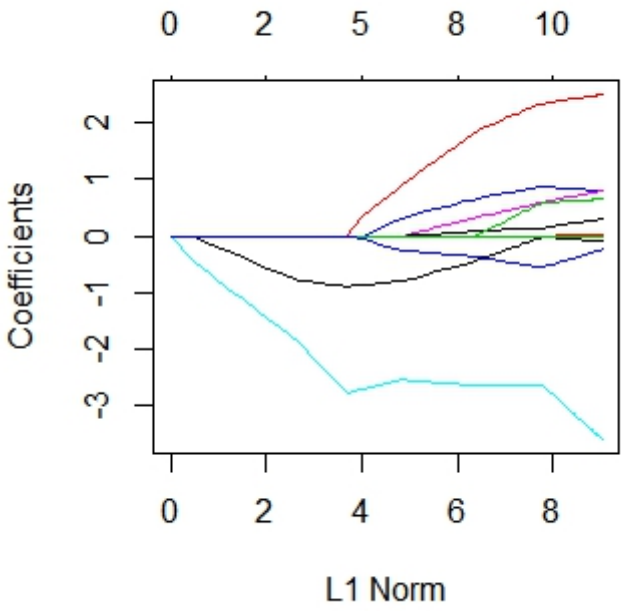

Figura 7.33: Elastic Net para $\alpha = 1$

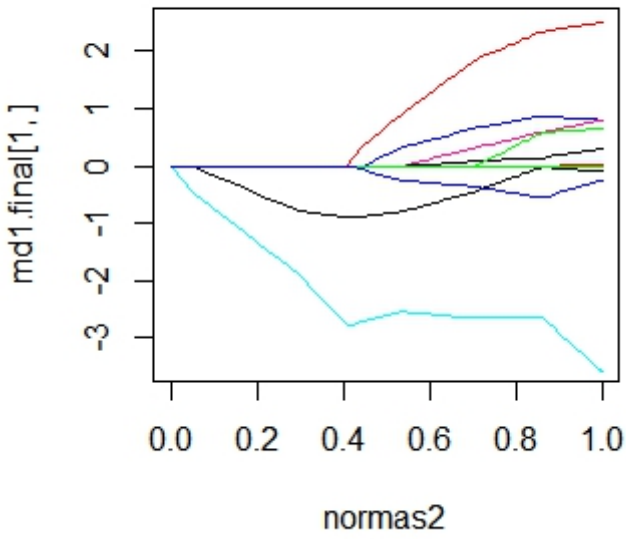

Figura 7.34: Gráfica del ajuste por LASSO una vez hechos los ajustes en los ejes

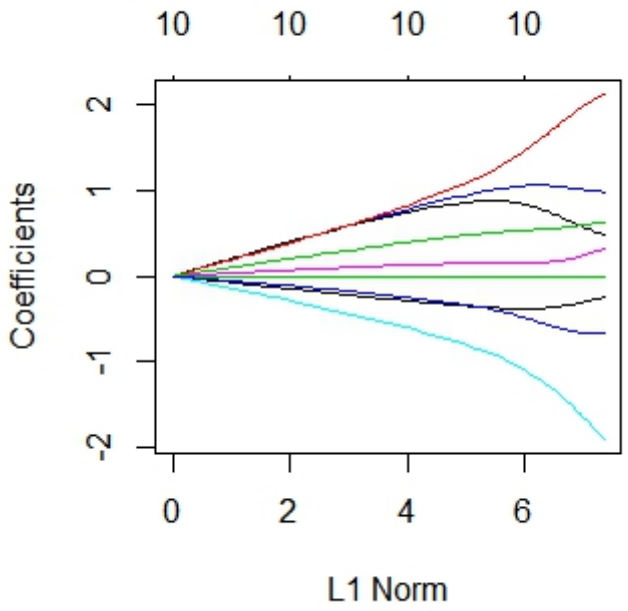

Figura 7.35: Elastic Net para $\alpha = 0$

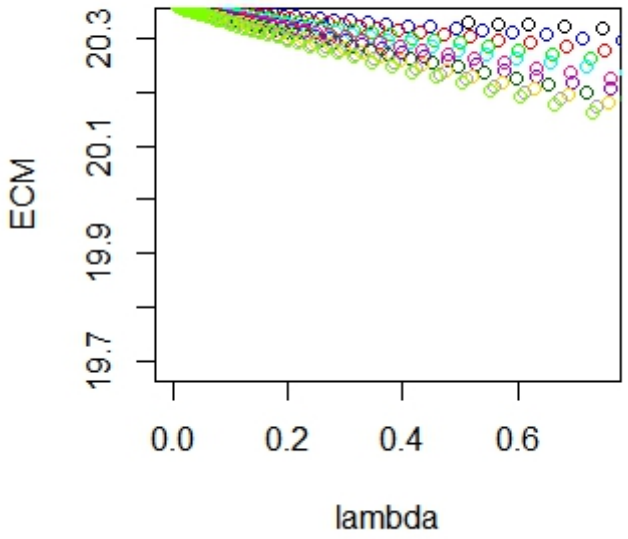

Figura 7.36: ECM frente a λ para cada valor de la partición de α

Bibliografía

[1] Trevor Hastie and Brad Efron. Paquete lars en r. urlhttps://CRAN.R-project.org/package=lars , 2013.

[2] C. M. Shetty Mokhtar S. Bazaraa, Hanif D. Sherali. Nonlinear programming: Theory and algorithms. *Wiley-Interscience*, 2(1):165–235, 2006.

[3] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[4] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):61–93, 1996.

[5] Lukas Meier Tim Hestergerg, Nam Hee Choi and Chris Fraley. Least angle and ℓ_1 penalized regression: A review. *Statistics Surveys*, 2(1):267–288, 2008.

[6] Jerome Friedman Trevor Hastie, Robert Tibshirani. *The Elements of Statistical Learning*. Springer, 2015.

[7] Robert Tibshirani Trevor Hastie and Jerome Friedman. Base de datos prostata. urlhttps://web.stanford.edu/ hastie/ElemStatLearn/ , 2013.

[8] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Royal Statistical Society*, 67(2):301–320, 2005.