

Laboratorio: Modelos de regresión lineal con R (III)

Jose Ameijeiras Alonso

1	Regresión lineal con regularización	1
1.1	Regresión Ridge	1
1.2	Regresión Lasso (least absolute shrinkage and selection operator)	3
1.3	Selección del parámetro de penalización	4

1 Regresión lineal con regularización

1.1 Regresión Ridge

Son varios los paquetes disponibles en R para el ajuste del modelo de regresión lineal mediante técnicas de regularización. Por ejemplo, para el caso de la regresión Ridge podremos usar, entre otros, los paquetes `MASS` y `glmnet`.

Trabajaremos con los datos discutidos en la sesión de teoría correspondientes al estudio sobre cáncer de próstata de Stamey et al. (1989). Los datos se encuentran disponibles en la librería `Brq`.

```
> library(Brq) # Prostate data set
> data(Prostate)
```

Los datos aparecen recogidos en un `data.frame` con 9 variables y 97 observaciones. El objetivo del estudio es determinar qué variables influyen en la presencia de un antígeno prostático específico (variable `lpsa`) para detectar el cáncer de próstata. Disponemos así como variables explicativas de medidas como el volumen del tumor (`lcavol`), el log-peso de la próstata (`lweight`), la edad (`age`), la cantidad de hiperplasia prostática benigna (`lbph`), el grado de infiltración del tumor en la vesícula seminal (`svi`), el grado de penetración capsular (`lcp`) y el score de Gleason (`gleason` y `pgg45`).

Proponemos un modelo de regresión lineal para explicar la variable `lpsa` en función del resto de marcadores clínicos ($p = 8$).

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p + \epsilon$$

En primer lugar, selecciona las observaciones correspondientes a la muestra de entrenamiento.

Ajusta un modelo de regresión lineal que explique la variable `lpsa` en función del resto de variables del modelo, utilizando el método de estimación de mínimos cuadrados (OLS). Recuerda que estimador por el método de mínimos cuadrados de los parámetros $\beta = (\beta_0, \beta_1, \dots, \beta_p)^t$ se obtienen resolviendo el problema de optimización:

$$\underset{\beta}{\text{Minimizar}} \quad \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2$$

A continuación se muestran los valores de los parámetros estimados.

	OLS
intercept	0.6694
lcavol	0.5870
lweight	0.4545
age	-0.0196
lbph	0.1071
svi	0.7662
lcp	-0.1055
gleason	0.0451
pgg45	0.0045

Ajustaremos ahora el mismo modelo de regresión lineal, mediante el procedimiento de estimación Ridge. Recuerda que en ese caso, los parámetros $\beta = (\beta_0, \beta_1, \dots, \beta_p)^t$ se obtienen resolviendo el problema de optimización

$$\underset{\beta}{\text{Minimizar}} \quad \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2 + \lambda \sum_{j=1}^p \beta_j^2.$$

La solución del problema depende del valor de λ seleccionado. En particular, para $\lambda = 0$ el problema se reduce al estimador ordinario de mínimos cuadrados. Para realizar el ajuste mediante el procedimiento de estimación Ridge con R, usamos la función `lm.ridge` del paquete `MASS`. Observa que la sintaxis es similar a la de la función `lm`, añadiendo el argumento `lambda` con el valor de la penalización λ que queramos considerar.

```
> library(MASS)
> rr <- lm.ridge(y ~ x, lambda = 0)
```

En cuanto a los parámetros estimados, debemos distinguir entre el resultado almacenado en `rr$coef` y los valores que nos devuelve `coef(rr)`. Puedes observar que el resultado de `coef(rr)` coincide con el obtenido por el método de mínimos cuadrados. La diferencia con respecto a los valores almacenados en `rr$coef` es que éstos no están en la escala original. Son los coeficientes que se obtienen de ajustar el modelo de regresión a las variables predictoras estandarizadas. Es decir, coincidirían con los resultados de ajustar el modelo de regresión lineal por el método de mínimos cuadrados a los datos que se obtienen tras restar a cada columna de la matriz `x` su media y dividir entre la desviación típica. Para comprobarlo, puedes utilizar la función `scale`, que estandariza una matriz de datos por columnas.

A continuación realizamos el ajuste mediante el procedimiento de estimación Ridge, usando diferentes valores del parámetro λ . Recuerda que el incremento del valor de λ implica la contracción del vector $(\hat{\beta}_1, \dots, \hat{\beta}_p)^t$.

```
> lam <- seq(0, 10, by = 0.01)
> rr <- lm.ridge(y ~ x, lambda = lam)
```

Puedes observar que ahora `coef(rr)` nos devuelve una matriz donde las filas recogen los parámetros estimados para cada valor de λ .

Otro de los paquetes de R que nos permiten realizar un ajuste lineal mediante el procedimiento de estimación Ridge es el paquete `glmnet`. El paquete `glmnet` es uno de los más utilizados en el contexto de regresión regularizada, ya que implementa en una única función (la función `glmnet`) distintos tipos de penalización. Veremos como es el uso de la función `glmnet` para regresión Ridge.

```
> library(glmnet)
> rr_glmnet <- glmnet(x, y, alpha = 0, lambda = 0)
```

El argumento `alpha` nos permite especificar el tipo de regularización que deseamos usar (`alpha=0` para Ridge). De nuevo, el argumento `lambda` corresponde al parámetro de penalización. Puedes observar con `coef(rr_glmnet)` que, como era de esperar, volvemos a obtener los mismos coeficientes estimados que con el ajuste por mínimos cuadrados. Usando diferentes valores del parámetro λ :

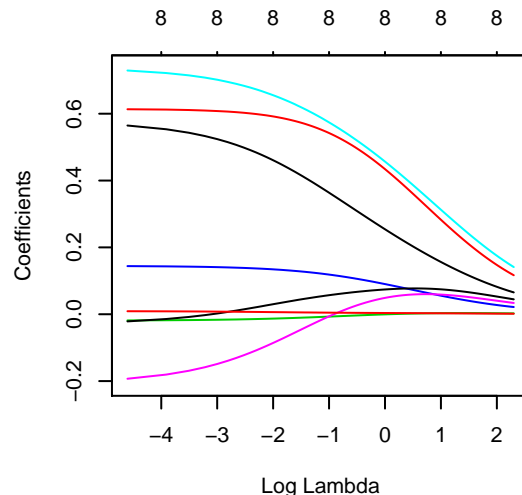
```
> lam <- seq(0, 10, by = 0.01)
> rr_glmnet <- glmnet(x, y, alpha = 0, lambda = lam)
```

Ahora, `coef(rr_glmnet)` nos devuelve una matriz donde las columnas recogen los parámetros estimados para cada valor de λ (las estimaciones se devuelven en la escala original). Fíjate que `coef(rr_glmnet)[,j]` son las estimaciones para el valor de λ almacenado en `rr_glmnet$lambda[j]`. Además, es importante mencionar que el problema de optimización que resuelve la función `glmnet` para Ridge es:

$$\underset{\beta}{\text{Minimizar}} \quad \frac{1}{2n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2 + \frac{\lambda}{2} \sum_{j=1}^p \beta_j^2.$$

Observa que no coincide exactamente con la formulación original de problema y esto hace que un mismo valor de $\lambda \neq 0$ no devuelva los mismos estimadores con la función `lm.ridge` y `glmnet`. Aún así el papel de λ como parámetro de penalización es el mismo, como se observa en la siguiente gráfica, que muestra las estimaciones de los parámetros a medida que aumenta λ .

```
> plot(rr_glmnet, xvar = "lambda")
```



1.2 Regresión Lasso (least absolute shrinkage and selection operator)

Al igual que ocurre con la regularización Ridge, son varios los paquetes de R que implementan el método de regularización Lasso. Nosotros realizaremos el ajuste con la función `glmnet` que, como hemos dicho, implementa distintos tipos de penalización dependiendo del valor del argumento `alpha` que seleccionemos. Antes de realizar el ajuste, recordamos que en este caso los parámetros $\beta = (\beta_0, \beta_1, \dots, \beta_p)^t$ se obtienen resolviendo el problema de optimización

$$\underset{\beta}{\text{Minimizar}} \quad \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2 + \lambda \sum_{j=1}^p |\beta_j|.$$

Ajustamos el modelo con la función `glmnet`

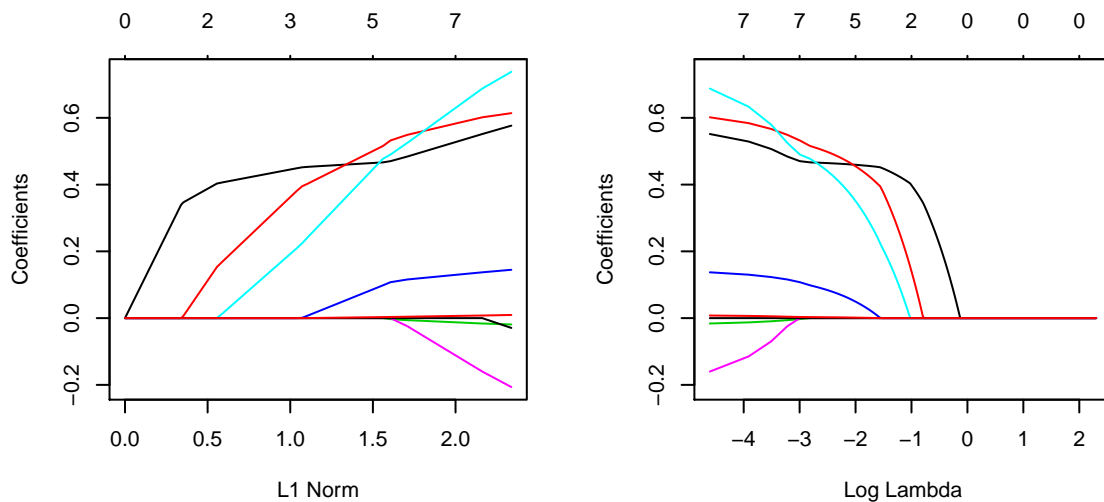
```
> library(glmnet)
> lasso <- glmnet(x, y, alpha = 1, lambda = 0)
```

El argumento `alpha` nos permite especificar el tipo de regularización que deseamos usar (`alpha=1` para Lasso). Al igual que antes, el argumento `lambda` corresponde al parámetro de penalización y, de nuevo, `coef(lasso)` nos devuelve los coeficientes estimados que coinciden con los obtenidos por el ajuste por mínimos cuadrados. Usando diferentes valores del parámetro λ :

```
> lam <- seq(0, 10, by = 0.01)
> lasso <- glmnet(x, y, alpha = 1, lambda = lam)
```

Ahora, `coef(lasso)` nos devuelve una matriz donde las columnas recogen los parámetros estimados para cada valor de λ (las estimaciones se devuelven en la escala original)¹. Representamos las estimaciones obtenidas en función de la norma L_1 del vector de parámetros estimado y en función del logaritmo de λ :

```
> plot(lasso)
> plot(lasso, xvar = "lambda")
```



1.3 Selección del parámetro de penalización

Tanto el método de regularización Ridge como el Lasso, requieren un método para la selección del parámetro de penalización λ adecuado. Uno de los métodos más utilizados para este propósito es la *validación cruzada*. El proceso fija un rango de posibles valores λ y ajusta el modelo para cada λ utilizando una parte de la muestra de entrenamiento (un subconjunto de observaciones se apartó previamente). A continuación se analiza la capacidad predictiva del modelo sobre el subconjunto de observaciones apartado (muestra de validación). Para ello es habitual calcular el error cuadrático medio (MSE) en la muestra de validación, es decir, $\frac{1}{n_v} \sum_{i=1}^{n_v} (y_i - \hat{y}_i)^2$, donde n_v es el tamaño de la muestra de validación, y_i son las observaciones de la muestra de validación y \hat{y}_i son las predicciones dadas por el modelo ajustado. El λ óptimo es aquel para el cual el MSE sea menor.

¹La función objetivo que minimiza `glmnet` para Lasso es $\frac{1}{2n} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2 + \lambda \sum_{j=1}^p |\beta_j|$.

Como este método depende en gran medida de qué observaciones se incluyen en la muestra para ajustar los modelos y cuáles en la muestra de validación, se han desarrollado otros métodos de validación cruzada.

Por ejemplo, la validación cruzada de k iteraciones (*k-fold cross-validation*) divide la muestra de entrenamiento en k subconjuntos de tamaño similar. En la primera etapa, se utiliza el primer subconjunto como muestra de validación, se ajusta el modelo en los $k - 1$ subconjuntos restantes y se calcula MSE_1 , el error cuadrático medio en la muestra de validación. A continuación se utiliza el segundo subconjunto como muestra de validación, se ajusta el modelo en los $k - 1$ subconjuntos restantes y se calcula MSE_2 , el error cuadrático medio en la muestra de validación. Por último, una vez repetido el mismo proceso para los k subconjuntos, se calcula la media aritmética de los MSE_i , con $i = 1, \dots, k$. El error obtenido se denomina error de validación cruzada y el λ óptimo es aquel para el cual el error de validación cruzada sea menor. Un caso particular de validación cruzada de k iteraciones es $k = n$ (*Leave-one-out cross-validation*), donde las muestras de validación en cada iteración están formadas por una única observación.

La función `cv.glmnet` realiza la validación cruzada de k iteraciones para `glmnet` y nos devuelve el valor óptimo de λ según este criterio. Veamos como se ejecuta en primer lugar para la regularización Ridge.

```
> lam <- seq(0, 10, by = 0.01)
> cvout <- cv.glmnet(x, y, alpha = 0, lambda = lam)
```

La función `cv.glmnet` usa por defecto $k = 10$. El valor λ que minimiza el error de validación cruzada se encuentra almacenado en `cvout$lambda.min` y los parámetros obtenidos al ajustar el modelo de regresión con regularización Ridge para ese valor de λ se obtienen directamente con la función `coef` como se muestra a continuación

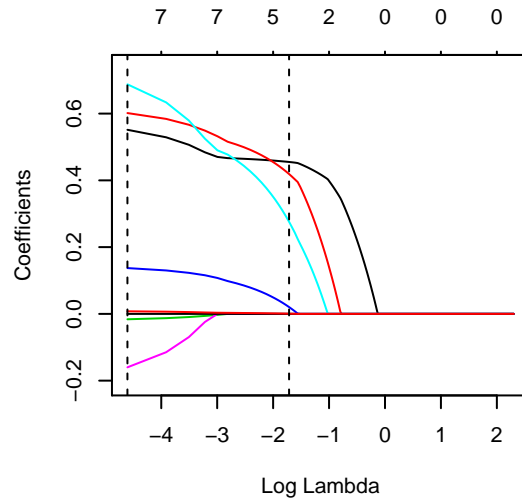
```
> cvout$lambda.min
> coef(cvout, s = "lambda.min")
```

La función también devuelve el valor `cvout$lambda.1se`, que representa un valor de λ para un modelo más simple que el de `cvout$lambda.min`, pero cuyo error está a una desviación estándar del mejor.

Del mismo modo, si queremos seleccionar el parámetro λ en el ajuste Lasso, haremos:

```
> lam <- seq(0, 10, by = 0.01)
> cvoutl <- cv.glmnet(x, y, alpha = 1, lambda = lam)
> cvoutl$lambda.min
> coef(cvoutl, s = "lambda.min")
> cvoutl$lambda.1se
> coef(cvoutl, s = "lambda.1se")
```

Mostramos a continuación una gráfica de los parámetros estimados con Lasso junto con el valor de λ seleccionado por validación cruzada (el valor óptimo `cvoutl$lambda.min` y `cvoutl$lambda.1se`)



En resumen, si seleccionamos como valores de λ los dados por `cvout$lambda.1se` (Ridge) y `cvout1$lambda.1se` (Lasso), los parámetros estimados con mínimos cuadrados, Ridge y Lasso son los que se muestran en la siguiente tabla:

	OLS	Ridge	Lasso
Intercept	0.6694	0.3470	1.2882
lcavol	0.5870	0.2803	0.4596
lweight	0.4545	0.3183	0.1369
age	-0.0196	-0.0022	0
lbph	0.1071	0.0555	0
svi	0.7662	0.4671	0.3216
lcp	-0.1055	0.0762	0
gleason	0.0451	0.0848	0
pgg45	0.0045	0.0027	0