Regularización y selección de características

Edmond Géraud

Selección de características

La selección de características es una técnica utilizada en aprendizaje automático para seleccionar un subconjunto de características relevantes y útiles para la tarea de predicción, con el objetivo de mejorar el rendimiento del modelo y reducir el riesgo de sobreajuste. En otras palabras, la selección de características se refiere al proceso de elegir un conjunto óptimo de características (variables o atributos) para el modelo predictivo, eliminando las características redundantes o irrelevantes que pueden afectar negativamente el rendimiento del modelo.

Selección de modelos por AIC

El criterio de información de Akaike (AIC) es un método para seleccionar modelos estadísticos entre un conjunto de modelos candidatos. Fue propuesto por el estadístico japonés Hirotugu Akaike en 1974.

El principio subyacente del AIC es que se prefiere un modelo que tenga un buen ajuste a los datos, pero que tenga un número mínimo de parámetros. El AIC combina estas dos medidas al considerar tanto la bondad de ajuste del modelo como el número de parámetros del modelo.

El AIC se define como la suma del error cuadrático (o alguna otra medida de error) del modelo y el producto del número de parámetros del modelo y una constante que depende del número de observaciones. El modelo con el valor más bajo del AIC se considera el mejor modelo de entre los modelos candidatos.

El AIC se basa en la idea de que el modelo más probable es aquel que minimiza la información perdida al modelar los datos, es decir, que tiene el equilibrio óptimo entre el ajuste y la complejidad del modelo. El término de penalización del número de parámetros en el AIC evita que el modelo se sobreajuste a los datos y selecciona el modelo más simple posible que aún pueda explicar los datos de manera efectiva.

En resumen, el principio subyacente del AIC es encontrar un modelo que tenga un buen ajuste a los datos, pero que tenga un número mínimo de parámetros. El AIC se basa en la idea de que el modelo más probable es aquel que minimiza la información perdida al modelar los

datos, y utiliza una medida combinada de la bondad de ajuste y la complejidad del modelo para seleccionar el mejor modelo de entre los modelos candidatos.

$$AIC = 2k - 2ln(L)$$

Donde k es el número de parámetros en el modelo y L es la función de verosimilitud máxima del modelo estimada a partir de los datos.

El AIC se calcula como la suma de dos términos: el primer término, 2k, es una penalización por el número de parámetros en el modelo, mientras que el segundo término, $-2ln(L)$, es proporcional al negativo del logaritmo de la función de verosimilitud máxima del modelo. El modelo con el valor más bajo del AIC se considera el mejor modelo de entre los modelos candidatos.

Métodos por reducción de dimensionalidad

PCA-PCR

El PCA (Principal Component Analysis) es una técnica estadística utilizada para reducir la dimensionalidad de los datos. El objetivo del PCA es encontrar una combinación lineal de variables predictoras (conocidas como componentes principales) que expliquen la mayor cantidad posible de la variación en los datos.

En el PCA, se asume que la variación en los datos se debe a una combinación de variables predictoras y no a variables aleatorias. Por lo tanto, el PCA busca identificar las variables predictoras que contribuyen más a la variación en los datos y combinarlas para formar nuevas variables o componentes principales.

La primera componente principal se elige de tal manera que tenga la mayor varianza posible, lo que significa que esta componente principal explica la mayor cantidad posible de la variación en los datos. Las componentes siguientes se eligen de tal manera que estén altamente correlacionadas con las variables predictoras originales, pero no estén correlacionadas con las componentes principales previamente seleccionadas.

Una vez que se han identificado las componentes principales, se pueden utilizar para reducir la dimensionalidad de los datos al proyectar los datos originales sobre las componentes principales seleccionadas. Esto se puede hacer eliminando las componentes principales que contribuyen menos a la variación en los datos y conservando solo las componentes principales más importantes.

El PCA se utiliza comúnmente en la exploración de datos y el análisis multivariado, particularmente en los casos en que hay muchas variables predictoras. También se utiliza en la clasificación y la agrupación de datos y en la visualización de datos en dos o tres dimensiones.

En resumen, el PCA es una técnica estadística utilizada para reducir la dimensionalidad de los datos. El PCA busca identificar las variables predictoras que contribuyen más a la variación en los datos y combinarlas para formar nuevas variables o componentes principales. Las componentes principales se pueden utilizar para reducir la dimensionalidad de los datos proyectando los datos originales sobre las componentes principales seleccionadas.

El PCR (Principal Component Regression) es un método de regresión que utiliza el PCA para reducir la dimensionalidad de los datos y luego realiza la regresión sobre las componentes principales seleccionadas. En lugar de utilizar todas las variables predictoras originales en la regresión, el PCR utiliza una combinación lineal de las componentes principales seleccionadas que mejor explique la variabilidad en la variable respuesta.

En el PCR, se realizan tres pasos principales. En primer lugar, se realiza el PCA sobre las variables predictoras para identificar las componentes principales que explican la mayor cantidad de variación en los datos. En segundo lugar, se seleccionan un número limitado de componentes principales que se utilizarán en la regresión. En tercer lugar, se realiza la regresión utilizando las componentes principales seleccionadas.

El objetivo del PCR es reducir la dimensionalidad de los datos para evitar el sobreajuste del modelo y mejorar la precisión de la regresión. El PCR se utiliza comúnmente en la regresión cuando hay muchas variables predictoras y es difícil identificar cuáles son las variables más importantes.

El PCR tiene algunas ventajas en comparación con otros métodos de regresión, como la regresión lineal múltiple. En particular, el PCR puede ser útil cuando hay una alta correlación entre las variables predictoras, lo que puede causar problemas en la regresión lineal múltiple. Además, el PCR puede mejorar la estabilidad de la regresión al reducir la dimensionalidad de los datos.

En resumen, el PCR es un método de regresión que utiliza el PCA para reducir la dimensionalidad de los datos y luego realiza la regresión sobre las componentes principales seleccionadas. El PCR se utiliza comúnmente en la regresión

Stop generating

PLS

El PLS (Partial Least Squares) es un método estadístico de regresión que busca establecer una relación lineal entre un conjunto de variables predictoras y una variable respuesta. El objetivo de PLS es encontrar una representación reducida de las variables predictoras, llamadas componentes latentes, que expliquen la mayor cantidad posible de variación en la variable respuesta.

La idea principal del PLS es encontrar una combinación lineal de las variables predictoras que explique la mayor parte de la variación en la variable respuesta. Para lograr esto, PLS utiliza

una técnica de descomposición en componentes principales que se enfoca en encontrar una relación lineal entre las variables predictoras y la variable respuesta.

La técnica de PLS busca encontrar una representación reducida de los datos de entrada mediante la creación de un conjunto de componentes latentes. Estos componentes latentes son combinaciones lineales de las variables predictoras originales que están altamente correlacionadas con la variable respuesta. Cada componente latente se construye de manera que tenga la máxima covarianza posible con la variable respuesta y, al mismo tiempo, esté altamente correlacionado con las variables predictoras originales.

La técnica de PLS se utiliza comúnmente en el análisis multivariado de datos, particularmente en la regresión de datos altamente correlacionados o cuando hay muchas variables predictoras en comparación con el número de observaciones disponibles. Además, PLS se puede utilizar para reducir la dimensionalidad de un conjunto de datos para su posterior análisis, así como para identificar las variables predictoras más importantes en un modelo de regresión.

Métodos de regularización

La regularización en los métodos de regresión es una técnica utilizada para evitar el sobreajuste o la falta de generalización de un modelo de regresión. El sobreajuste ocurre cuando un modelo se ajusta demasiado bien a los datos de entrenamiento, lo que puede hacer que sea demasiado específico y no se generalice bien a nuevos datos. La regularización es una técnica que impone restricciones al modelo de regresión, con el objetivo de reducir la complejidad y evitar el sobreajuste.

Existen diferentes tipos de regularización, como la regularización L1 (también conocida como LASSO), la regularización L2 (también conocida como Ridge) y la regularización Elastic Net, que combina ambas técnicas. Estas técnicas agregan una penalización a la función de costo del modelo de regresión, lo que permite controlar la complejidad del modelo y evitar el sobreajuste.

La regularización es una técnica muy útil en el aprendizaje automático, especialmente en problemas de regresión donde se trabaja con grandes conjuntos de datos y se desea obtener modelos que sean generalizables y no sobreajustados a los datos de entrenamiento.

La regularización L2 agrega una penalización a la función de costo del modelo de regresión que es proporcional al cuadrado de los coeficientes del modelo. Esta penalización reduce los coeficientes de las características menos relevantes, lo que significa que el modelo se centrará más en las características que son más importantes para la predicción.

Una vez que se ha entrenado el modelo de regresión con la técnica de regularización L2, es posible utilizar los coeficientes resultantes para identificar las características más importantes. Estas características pueden seleccionarse y utilizarse para entrenar otro modelo, como un modelo de clasificación, por ejemplo.

Es importante tener en cuenta que la selección de características debe realizarse con cuidado y que no siempre es adecuado utilizar solo las características más importantes para entrenar un modelo, ya que pueden perderse información importante. Por lo tanto, es recomendable realizar un análisis cuidadoso de las características antes de seleccionar las más relevantes.

Si el modelo de regresión lineal es el siguiente:

$$Y = X\beta + \epsilon$$

La regularización L1 o Lasso intenta minimizar la siguiente ecuación

$$\min \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

La regularización L2 o ridge:

$$\min \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

La elastic net:

$$\min \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

Los métodos de regularización como L1, L2 y Elastic Net no tienen una solución cerrada o analítica, y generalmente se utilizan métodos numéricos iterativos para encontrar los coeficientes óptimos del modelo.

Estos métodos iterativos se basan en el principio de minimización del error cuadrático (o alguna otra medida de error) del modelo, junto con la función de regularización. Por ejemplo, el algoritmo de descenso de gradiente puede ser utilizado para optimizar la función objetivo de regresión con regularización.

Además, es importante ajustar el parámetro de regularización λ (o λ_1 y λ_2 en el caso de Elastic Net) para encontrar un equilibrio adecuado entre el ajuste del modelo y la regularización. Esto se puede hacer mediante la validación cruzada o mediante el uso de algoritmos de optimización más avanzados, como la búsqueda de cuadrícula o la optimización bayesiana.

En resumen, se utilizan métodos iterativos y de optimización para encontrar la solución óptima en métodos de regularización, lo que hace que su implementación sea más compleja que en los métodos de regresión lineal sin regularización.

El método de descenso de gradiente es un algoritmo iterativo de optimización utilizado para minimizar una función de costo. En el contexto de los métodos de regresión con regularización, esta función de costo se define como la suma del error cuadrático (o alguna otra medida de error) del modelo, junto con la función de regularización.

El algoritmo de descenso de gradiente comienza con un valor inicial de los coeficientes del modelo, y en cada iteración actualiza los valores de los coeficientes en la dirección opuesta al gradiente de la función de costo. La idea es que al actualizar los coeficientes en esta dirección, se reducirá gradualmente el valor de la función de costo.

El tamaño de los pasos que se dan en cada iteración se controla mediante un parámetro conocido como la tasa de aprendizaje. Si la tasa de aprendizaje es demasiado pequeña, el algoritmo puede converger lentamente, mientras que si es demasiado grande, puede no converger en absoluto. Por lo tanto, elegir una tasa de aprendizaje adecuada es importante para el éxito del algoritmo.

El algoritmo de descenso de gradiente puede repetirse hasta que se alcanza un criterio de convergencia predefinido, como una tolerancia de error o un número máximo de iteraciones. Una vez que se alcanza la convergencia, se devuelve el valor óptimo de los coeficientes del modelo.

Métricas

Existen varias métricas que se utilizan comúnmente para evaluar la rendición de los modelos, dependiendo del tipo de problema y del tipo de modelo utilizado. Aquí te presento algunas de las métricas más comunes:

- Error cuadrático medio (MSE): Esta métrica mide la media de los errores al cuadrado entre las predicciones del modelo y los valores reales de la variable respuesta. El MSE es muy utilizado en problemas de regresión.
- Coeficiente de determinación (R^2): Esta métrica mide la proporción de la variabilidad en la variable respuesta que es explicada por el modelo. Un valor de R^2 cercano a 1 indica que el modelo explica bien la variabilidad en la variable respuesta, mientras que un valor cercano a 0 indica que el modelo no es capaz de explicar la variabilidad. El R^2 es muy utilizado en problemas de regresión.
- Exactitud (accuracy): Esta métrica mide la proporción de predicciones correctas del modelo en relación al total de predicciones. El accuracy es muy utilizado en problemas de clasificación.
- Precisión (precision) y recuperación (recall): Estas métricas se utilizan comúnmente en problemas de clasificación binaria para evaluar la calidad de las predicciones positivas del modelo. La precisión mide la proporción de predicciones positivas correctas en relación

al total de predicciones positivas, mientras que la recuperación mide la proporción de instancias positivas que son correctamente clasificadas por el modelo.

- F1-score: Esta métrica combina la precisión y la recuperación en una única medida que se utiliza comúnmente en problemas de clasificación binaria.

Cabe destacar que no hay una métrica única que sea la más utilizada para evaluar la rendición de los modelos, ya que la elección de la métrica depende del tipo de problema y del tipo de modelo utilizado. En general, se recomienda utilizar varias métricas para evaluar la rendición de los modelos y obtener una visión más completa del desempeño del modelo.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

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

Práctica

De los datos `fat` del paquete `faraway`,utiliza el porcentaje de grasa,`siri`, como la respuesta, y las otras variables como independientes, excepto `brozek` y `density`. Quita una observación cada 10 del conjunto de datos para formar el conjunto de entrenamiento y el de prueba.

Realiza los siguientes modelos:

1. **Regresión múltiple lineal**
2. **Regresión múlitple seleccionando variables con AIC**
3. **PCR**
4. **PLS**
5. **Ridge**
6. **LASSO**

Cargamos librerías y carga de datos

```
library(faraway)
library(ggstatsplot)
```

```
You can cite this package as:
     Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach.
     Journal of Open Source Software, 6(61), 3167, doi:10.21105/joss.03167
```

```r
library(DescTools)
library(corrplot)
```

```
corrplot 0.92 loaded
```

```r
library(leaps)
library(pls)
```

```
Attaching package: 'pls'

The following object is masked from 'package:corrplot':

    corrplot

The following object is masked from 'package:stats':

    loadings
```

```r
library(MASS)
library(glmnet)
```

```
Loading required package: Matrix

Loaded glmnet 4.1-7
```

```r
datos <- fat
str(fat)
```

```
'data.frame':   252 obs. of  18 variables:
 $ brozek : num  12.6 6.9 24.6 10.9 27.8 20.6 19 12.8 5.1 12 ...
 $ siri   : num  12.3 6.1 25.3 10.4 28.7 20.9 19.2 12.4 4.1 11.7 ...
 $ density: num  1.07 1.09 1.04 1.08 1.03 ...
 $ age    : int  23 22 22 26 24 24 26 25 25 23 ...
 $ weight : num  154 173 154 185 184 ...
 $ height : num  67.8 72.2 66.2 72.2 71.2 ...
 $ adipos : num  23.7 23.4 24.7 24.9 25.6 26.5 26.2 23.6 24.6 25.8 ...
 $ free   : num  135 161 116 165 133 ...
 $ neck   : num  36.2 38.5 34 37.4 34.4 39 36.4 37.8 38.1 42.1 ...
 $ chest  : num  93.1 93.6 95.8 101.8 97.3 ...
 $ abdom  : num  85.2 83 87.9 86.4 100 94.4 90.7 88.5 82.5 88.6 ...
 $ hip    : num  94.5 98.7 99.2 101.2 101.9 ...
 $ thigh  : num  59 58.7 59.6 60.1 63.2 66 58.4 60 62.9 63.1 ...
 $ knee   : num  37.3 37.3 38.9 37.3 42.2 42 38.3 39.4 38.3 41.7 ...
 $ ankle  : num  21.9 23.4 24 22.8 24 25.6 22.9 23.2 23.8 25 ...
 $ biceps : num  32 30.5 28.8 32.4 32.2 35.7 31.9 30.5 35.9 35.6 ...
 $ forearm: num  27.4 28.9 25.2 29.4 27.7 30.6 27.8 29 31.1 30 ...
 $ wrist  : num  17.1 18.2 16.6 18.2 17.7 18.8 17.7 18.8 18.2 19.2 ...
```

Estadística descriptiva e inferencial

Nos dice el enunciado que tenemos que quitar `brozek` y `density`

```
X <- datos[,-c(1,3)]
```

El conjunto de datos lo constan 252 observaciones y 18 variables. Todas las variables son numéricas

Realizamos pruebas de Jarque Bera para ver la normalidad de los datos

```
w <- which(apply(X,2,function(x) JarqueBeraTest(x)$p.value)<0.05)
length(w)
```

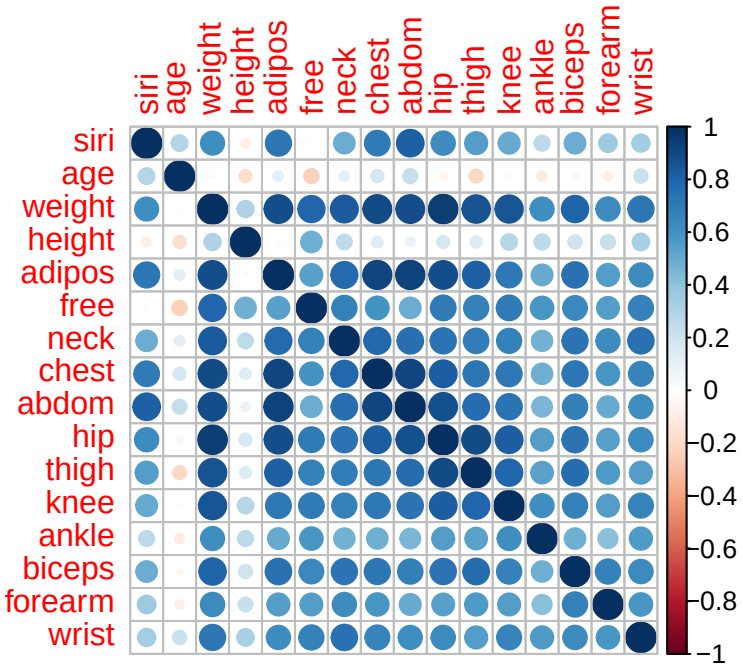
```
[1] 11
```

De las 16 variables, 11 no tienen normalidad.

Procedemos a realizar la correlación entre las variables.

Cómo tenemos variables que no siguen la normalidad tendríamos que hacer una prueba de **spearman**, pero como son bastantes observaciones la de **pearson** es más robusta

```
correlacion <- cor(X,method = "pearson")
corrplot::corrplot(correlacion)
```



Podemos observar, como muchas de las variables estan íntimamente correlacionadas, por lo tanto, a priori ya sabemos que nuestros modelos múltiples lineales simples, no serán buenos.

Antes de empezar necesitamos escalar los datos

```
X <- as.data.frame(scale(X))
```

Partición de datos prueba y entrenamiento

```
train <- X[-seq(10,252,10),]
test <- X[seq(10,252,10),]
```

Regresión lineal múltiple

```
g  <- lm(siri ~.,data=train)
summary(g)
```

```
Call:
lm(formula = siri ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.69681 -0.08032  0.02185  0.10933  0.79604

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.005917   0.012328  -0.480 0.631736
age          0.012014   0.018553   0.648 0.517983
weight       1.274772   0.081873  15.570  < 2e-16 ***
height       0.021458   0.017645   1.216 0.225315
adipos      -0.224077   0.049727  -4.506 1.09e-05 ***
free        -1.230381   0.032435 -37.933  < 2e-16 ***
neck         0.004800   0.026103   0.184 0.854272
chest        0.121106   0.039882   3.037 0.002694 **
abdom        0.180529   0.054356   3.321 0.001056 **
hip          0.005305   0.048026   0.110 0.912148
thigh        0.122365   0.034165   3.582 0.000424 ***
knee         0.030732   0.026956   1.140 0.255542
ankle        0.025340   0.016466   1.539 0.125325
biceps       0.034730   0.023342   1.488 0.138278
forearm      0.055722   0.017707   3.147 0.001888 **
wrist        0.015537   0.023070   0.673 0.501378
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1852 on 211 degrees of freedom
Multiple R-squared:  0.9692,    Adjusted R-squared:  0.967
F-statistic: 442.5 on 15 and 211 DF,  p-value: < 2.2e-16
```

Observamos, cómo 7 variables son las más importantes en el modelo. De hecho tenemos un R2 de casi el 97 porciento

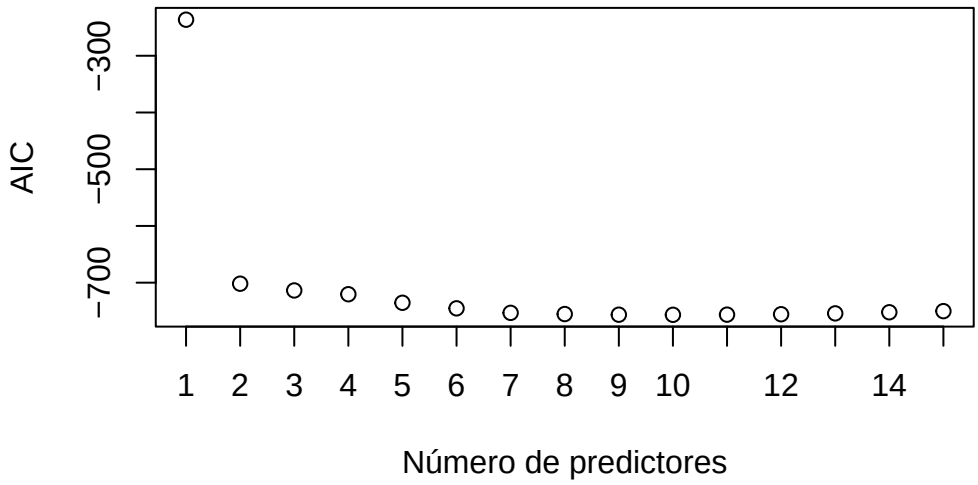
Predicción

```
predlm  <- predict(g,test)
```

Regresión lineal múltiple. Selección AIC

Lo podemos plantear manualmente

```
conj <- regsubsets(siri ~., data=train, nvmax = 15)
rconj <- summary(conj)
n <- nrow(train)
p <- ncol(train[,-1]) + 1
aic <- n*log(rconj$rss/n)+(2:p)*2
plot(1:(p-1),aic,ylab="AIC",xlab="Número de predictores",axes=F)
box(); axis(1,at=1:(p-1)); axis(2)
```



El mínimo AIC se obtiene con

```
which.min(aic)
```

```
[1] 10
```

pero con 7, 8 o 9 predictoras, el valor de AIC es similar.

Si se utiliza la función `step()`, se minimiza el AIC con un modelo de 10 predictores

```r
lmaic <- step(g, trace = F)
formula(lmaic)
```

```
siri ~ weight + adipos + free + chest + abdom + thigh + knee +
    ankle + biceps + forearm
```

```r
predaic <- predict(lmaic,test)
```

PCR

```r
mpc <- pcr(siri ~., data=train, validation="CV")
mpcCV <- RMSEP(mpc, estimate="CV")
(numpredcp <- which.min(mpcCV$val))
```

```
[1] 16
```

En este caso el número de predictores seleccionados es de 7 (descontando el intercept).

```r
predcp <- predict(mpc, test, ncomp=numpredcp-1)
```

Regresion por mínimos cuadrados parciales (PLS)

```r
set.seed(123456)
mpls <- plsr(siri~., data=train, validation="CV")
mplsCV <- RMSEP(mpls, estimate="CV")
(numpredpls<-which.min(mplsCV$val) - 1)
```

```
[1] 4
```

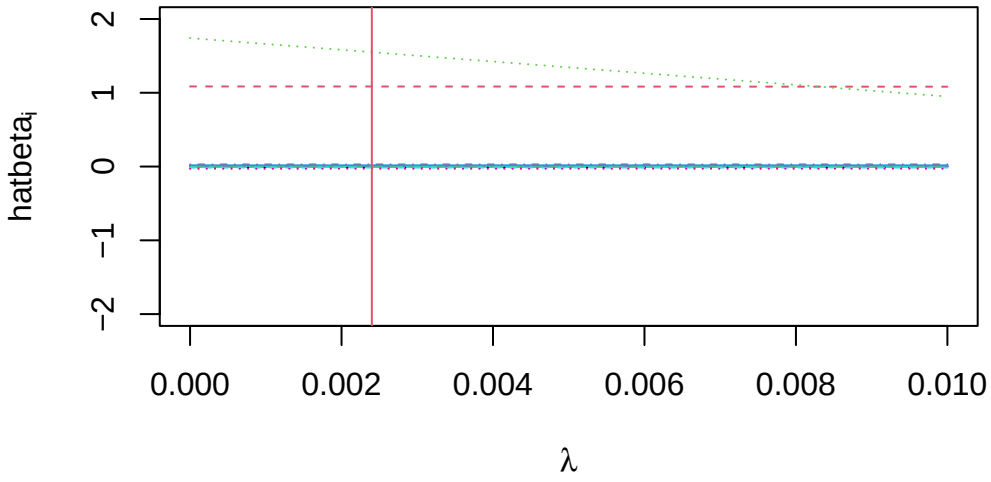
```r
predpls <- predict(mpls,test,ncomp=numpredpls)
```

Regresión contraída (RIDGE)

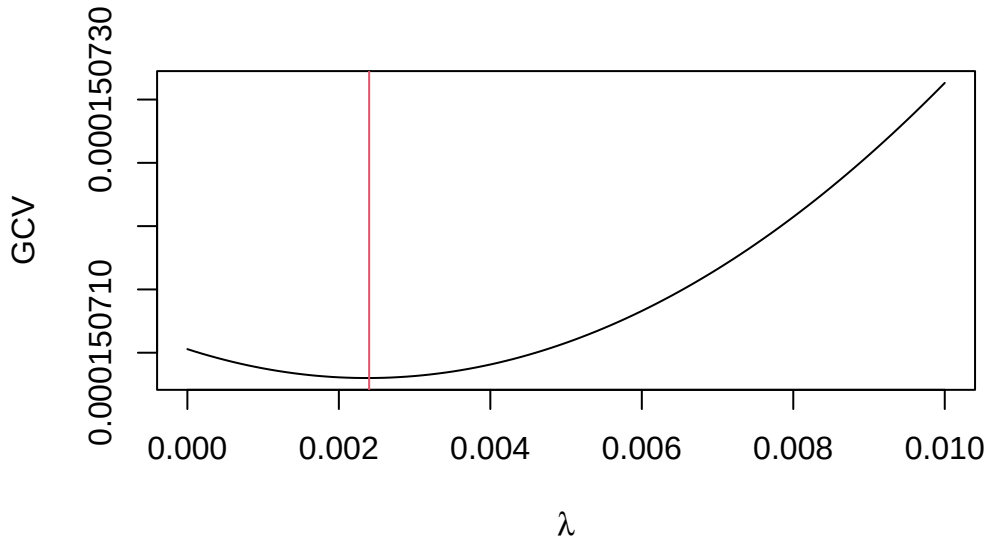
```
mr <- lm.ridge(siri ~., data=datos, lambda=(seq(0,0.01,0.0001)))
(nGCV <- which.min(mr$GCV))
```

```
0.0024
    25
```

```
set.seed(124568919)
lGCV <- mr$lambda[nGCV]
matplot(mr$lambda,coef(mr),type="l", ylim=c(-2,2),                xlab=expression(lambda)
abline(v=lGCV,col=2)
```




```
plot(mr$lambda,mr$GCV,type="l",xlab=expression(lambda),ylab="GCV")
abline(v=lGCV,col=2)
```



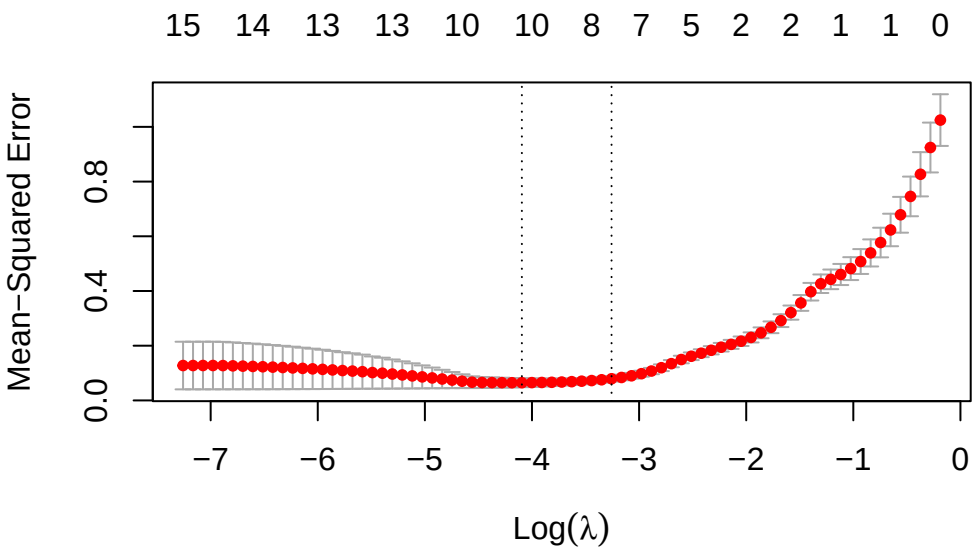

```
mr <- lm.ridge(siri~.,train, lambda=lGCV)
predridge <- cbind(1,as.matrix(test[,-1])) %*% coef(mr)
```

Regresión LASSO

```
#perform k-fold cross-validation to find optimal lambda value
train.matrix <- as.matrix(train)
X <- train.matrix[,-1]
y <- train.matrix[,1]
mod_cv <- cv.glmnet(x=X, y=y, family="gaussian",
                        intercept = F, alpha=1)
#find optimal lambda value that minimizes test MSE
best_lambda <- mod_cv$lambda.min
best_lambda
```

```
[1] 0.01666548
```

```
#produce plot of test MSE by lambda value
plot(mod_cv)
```




```
test.matrix <- as.matrix(test)
predlasso <- predict(mod_cv,test.matrix[,-1])
```

Resumen

```
 rmse <- function(x,y) sqrt(mean((x-y)^2))
 rmse.lm <- rmse(predlm,test$siri)
 rmse.aic <- rmse(predaic,test$siri)
 rmse.cp <- rmse(predcp,test$siri)
 rmse.pls <- rmse(predpls,test$siri)
 rmse.ridge <- rmse(predridge,test$siri)
 rmse.lasso <- rmse(predlasso,test$siri)

 res <- data.frame(lm=rmse.lm,
            aic=rmse.aic,
            pcr = rmse.cp,
            pls = rmse.pls,
            ridge=rmse.ridge,
            lasso=rmse.lasso)

 res <- as.data.frame(t(round(res,4)))
res
```

```
          V1
lm    0.1352
aic   0.1341
pcr   0.1352
pls   0.2012
ridge 0.1352
lasso 0.2015
```