Regresión logística

September 7, 2017

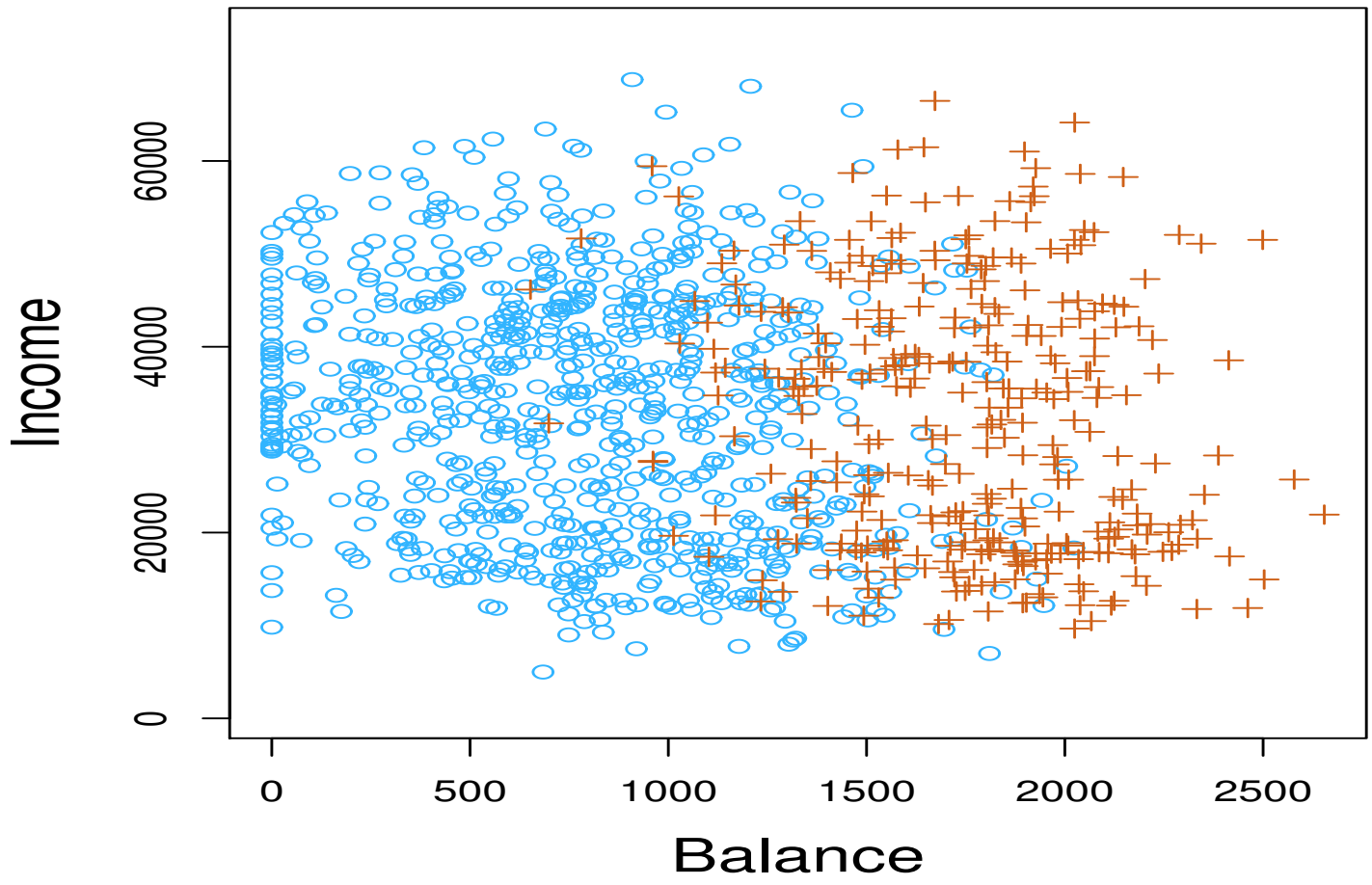
Contenidos

Regresión logística.

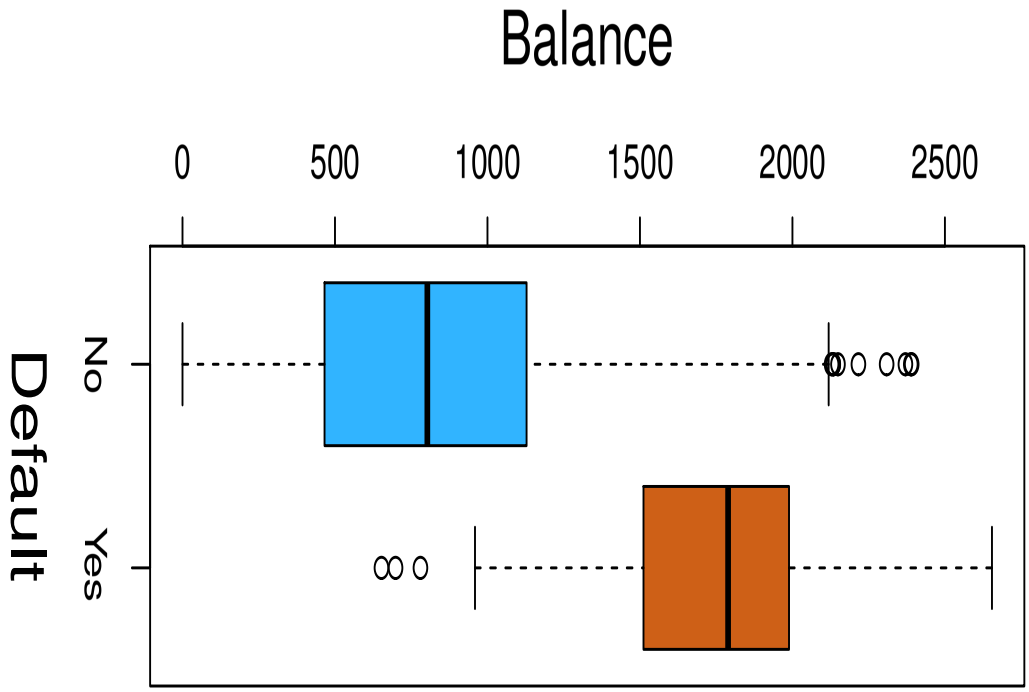
Regresión logística regularizada

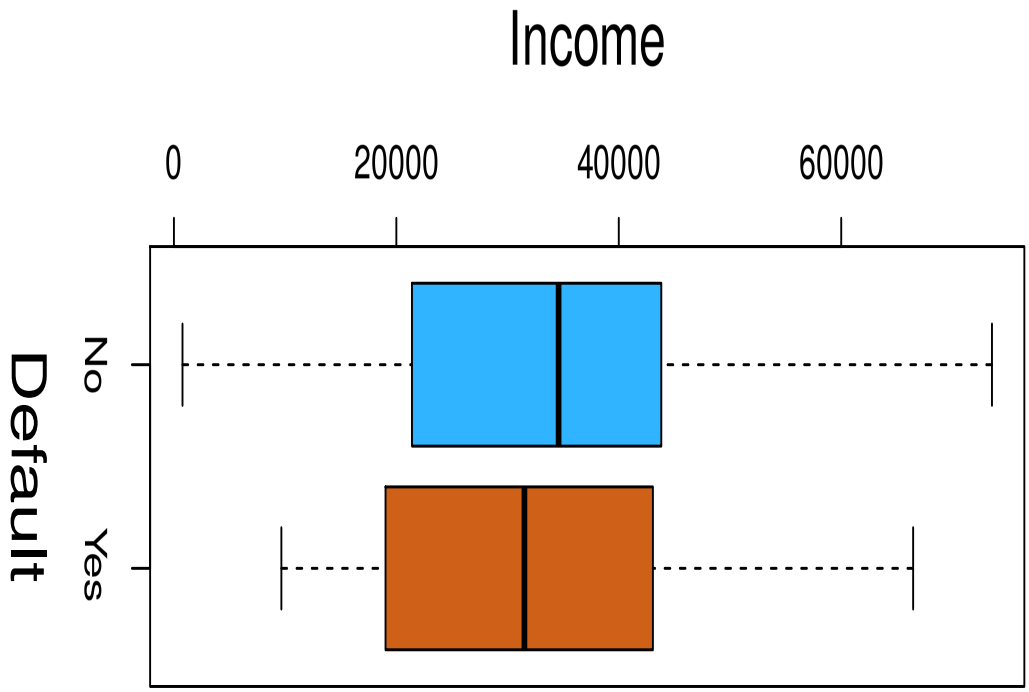
Datos de Default

Queremos predecir si un individuo entrará en cesación de pagos con su tarjeta de crédito usando las covariables:
`income`: ingreso anual
`balance`: balance anual
`student`: si el individuo es o no estudiante.



Ingreso anual versus balance promedio mensual de un conjunto de individuos. Los que entraron en default están representados en naranja.


Balance
0
500
1000
1500
2000
2500
Default
No
Yes


Income
0
20000
40000
60000
Default
No
Yes

```
> attach(Default)
> mean(default=="Yes")
[1] 0.0333
> mean(default[student=="Yes"]=="Yes")
[1] 0.04313859
> mean(default[student=="No"]=="Yes")
[1] 0.02919501
```

El modelo de regresión logística

$$P(Y = 1|X) = \text{expit}\,(\beta_0 + \beta_1 X_1 + \ldots \beta_p X_p)$$

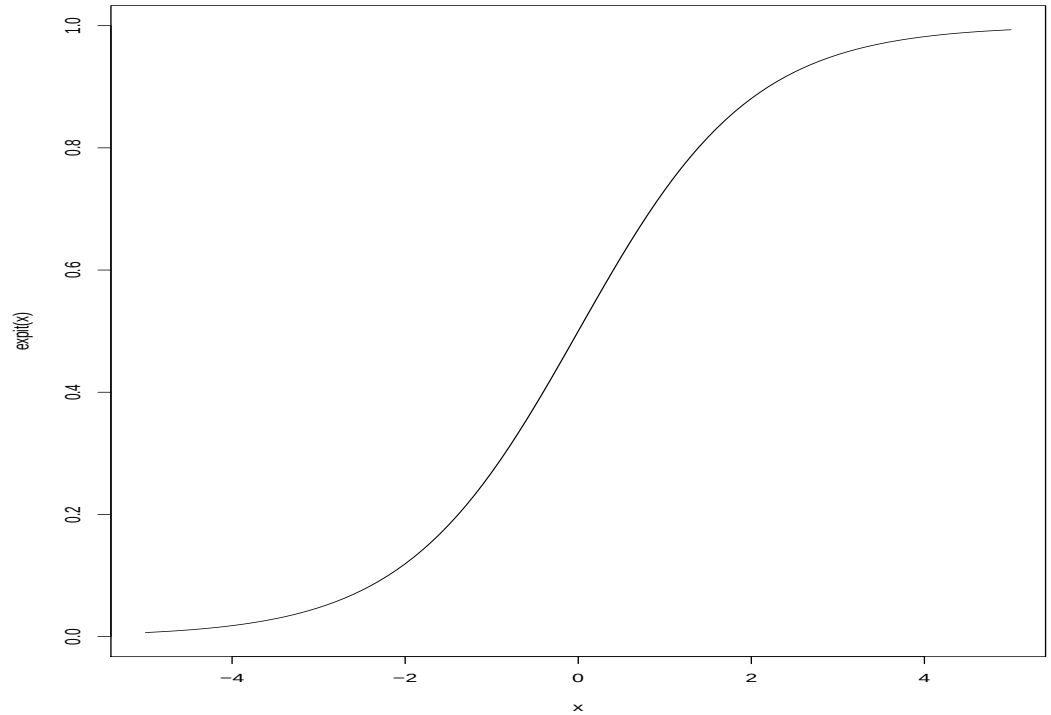
$$\text{expit}(t) = \frac{e^t}{1 + e^t}$$

o, equivalentemente,

$$\text{logit}(P(Y = 1|X)) = \beta_0 + \beta_1 X_1 + \ldots \beta_p X_p$$

donde

$$\text{logit}(t) = \log\left(\frac{t}{1 - t}\right)$$


expit(x)
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
x

La razón de probabilidades u *odds*

El modelo de regresión logística puede escribirse como

$$\frac{P(Y=1|X)}{1-P(Y=1|X)} = e^{\beta_0+\beta_1 X_1+\ldots\beta_p X_p}$$

El valor $P(Y=1|X)/(1-P(Y=1|X))$ de llama razón de probabilidades u *odds* y toma valores entre 0 y $+\infty$.
Por ejemplo, una probabilidad de caer en default de $p=0.2$ resulta en un odds de $0.2/0.8=1/4$.
Interpretación:
De cada 100 personas con este odds, aproximadamente 20 entrarán en default y 80 no.
Por cada persona con este odds que cae en default hay 4 que no.

Estimación

Los estimadores de $\beta_0, \beta_1, \ldots, \beta_p$ se encuentra por máxima verosimilitud, maximizando

$$L(\beta_0, \beta_1, \ldots, \beta_p) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1-y_i},$$

o bien

$$\log L(\beta_0, \beta_1, \ldots, \beta_p) = \sum_{i=1}^{n} y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i)$$

donde

$$\pi_i = \text{expit}\left(\beta_0 + \beta_1 X_1 + \ldots \beta_p X_p\right).$$

Ajuste para los datos de default

```
> defaultglm<-glm(default~student+balance+income,data=Default,
family="binomial")
> summary(defaultglm)

Call:
glm(formula = default ~ student + balance + income,
family = "binomial",
    data = Default)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4691  -0.1418  -0.0557  -0.0203   3.7383

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619
balance      5.737e-03  2.319e-04  24.738  < 2e-16
income       3.033e-06  8.203e-06   0.370  0.71152
---
```

Bajo los supuestos del modelo logístico, para n grande,

$$\frac{\hat{\beta}_i - \beta_i}{\hat{SE}(\hat{\beta}_i)} \sim \mathcal{N}(0,1) \text{ aproximadamente}$$

Sea $H_0 : \beta_i = 0$ vs $H_0 : \beta_i \neq 0$

A nivel 0.05, se rechaza H_0 si $\frac{\hat{\beta}_i}{\hat{SE}(\hat{\beta}_i)} > 1.96$

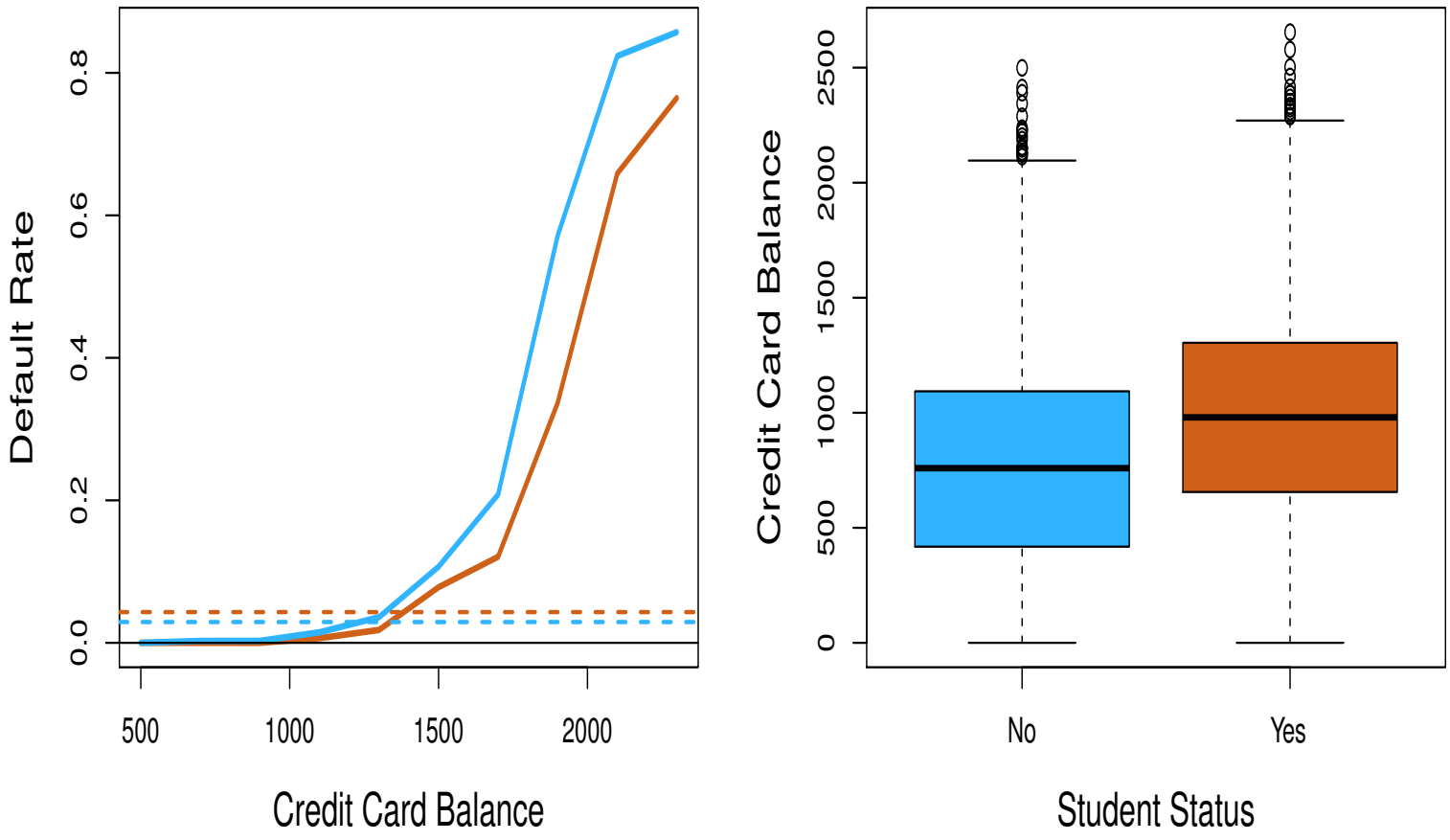
Interpretación de los coeficientes

Tener balance alto aumenta la probabildad de entrar en default, si las variables `student` e `income` permanecen fijas.

Ser estudiante disminuye la probabildad de entrar en default, si las variables `balance` e `income` permanecen fijas.

El nivel de ingresos no tiene efecto significativo en la probabildad de entrar en default una vez que se han considerado `balance` e `student`.

¿Esto contradice el hecho de que la proporción de personas que entraron en default es mayor entre los estudiantes que entre los no estudiantes?



La curva y la caja naranjas corresponden a los estudiantes; las azules a los no estudiantes.

Predicciones

Una vez estimados los coeficientes podemos estimar la probabilidad de un individuo de entrar en default. Por ejemplo, si un estudiante tiene un ingreso anual de 22500 dólares y un balance promedio de 1400 dólares, su probabilidad de entrar en default es:

$$\frac{e^{-10.87-0.6468+0.0057*1400+0.000003033*22500}}{1+e^{-10.87-0.6468+0.0057*1400+0.000003033*22500}} = 0.487863$$

para un no estudiante con el mismo ingreso y balance, la probabilidad de entrar en default es:

$$\frac{e^{-10.87+0.0057*1400+0.000003033*22500}}{1+e^{-10.87+0.0057*1400+0.000003033*22500}} = 0.6452541$$

El banco clasificará como clientes de alto riesgo a aquellos que tengan una probabilidad de entrar en default mayor a 0.5 (podría ser más exigente y poner como límite 0.1).

El estudiante mencionado será clasificado como cliente de bajo riesgo, mientras que el no estudiante será clasificado como cliente de alto riesgo.

Error de clasificación

Hay dos tipos:
Error de clasificación de entrenamiento: Es la proporción de observaciones de la muestra de entrenamiento que resultan mal clasificadas.

Error de clasificación de testeo: Es la probabilidad de que una nueva observación resulte mal clasificada.

Error de clasificación

Hay dos tipos:
Error de clasificación de entrenamiento: Es la proporción de observaciones de la muestra de entrenamiento que resultan mal clasificadas.

Error de clasificación de testeo: Es la probabilidad de que una nueva observación resulte mal clasificada.

Calculando el error de entrenamiento

```
> predprobs<-predict(defaultglm,type="response")
> head(predprobs)
           1            2            3            4
0.0014287239 0.0011222039 0.0098122716 0.0004415893
           5            6
0.0019355062 0.0019895182
> predclas<-predprobs>=0.5
> head(predclas)
    1     2     3     4     5     6
FALSE FALSE FALSE FALSE FALSE FALSE
```

```
> mat<-table(predclas,default)
> mat
         default
predclas   No  Yes
   FALSE 9627  228
   TRUE    40  105
> bienclas<-sum(diag(mat))
> bienclas
[1] 9732
> total<-length(default)
> malclas<-total-bienclas
> malclas
[1] 268
> errdeclas<-malclas/total
> errdeclas
[1] 0.0268
```

Estimando el error de testeo

```
> set.seed(10)
> n<-nrow(Default)
> train<-sample(n,7000)
> test<-(-train)
> defaultglmtrain<-glm(default~student+balance+income,
data=Default[train,],family="binomial")
> predprobs<-predict(defaultglm,type="response",
newdata=Default[test,])
> predclas<-predprobs>=0.5
> mat<-table(predclas,default[test])
> mat

predclas    No  Yes
   FALSE 2883   75
   TRUE    13   29
```

```
> bienclas<-sum(diag(mat))
> bienclas
[1] 2912
> total<-length(default[test])
> malclas<-total-bienclas
> malclas
[1] 88
> errdeclas<-malclas/total
> errdeclas
[1] 0.02933333
```

Métodos robustos para modelos lineales generalizados

Para calcular estimadores robustos para modelos lineales generalizados:

```
library(robustbase)
defaultglmrob<-glmrob(default~student+balance+income,
data=Default,
family="binomial")
```

Hay varios métodos robustos. El que hace el R por default es el método "Mqle", propuesto en Cantoni y Ronchetti (2001).

Nosotros preferimos los métodos:
"BY" para regresión logística (propuesto por Bianco y Yohai (1996)).

"MT" para regresión de Poisson (propuesto por Valdora y Yohai (2014)).

En los datos de default obtenemos

```
> summary(defaultglmrob)

Call:glmrob(formula = default ~ student+balance+income,
 family = "binomial", method = "BY")


Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.098e+01  5.680e-01 -19.335   <2e-16
studentYes  -6.535e-01  2.674e-01  -2.444   0.0145
balance      5.796e-03  2.669e-04  21.715   <2e-16
income       3.065e-06  9.306e-06   0.329   0.7419
```

El ajuste da muy parecido al ajuste por máxima verosimilitud. Esto indica que no hay outliers de los cuales preocuparse.

Ejemplo: datos de expresión genética para clasificación de leucemia

En el tratamiento del cancer es muy importante clasificar los tumores para maximizar la eficiencia y minimizar la toxicidad.

Ejemplo: datos de expresión genética para clasificación de leucemia

En el tratamiento del cancer es muy importante clasificar los tumores para maximizar la eficiencia y minimizar la toxicidad.

Analizamos un conjunto de datos que fue introducido en Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., ...& Bloomfield, C. D. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science, 286(5439), 531-537.

Ejemplo: datos de expresión genética para clasificación de leucemia

En el tratamiento del cancer es muy importante clasificar los tumores para maximizar la eficiencia y minimizar la toxicidad.

Analizamos un conjunto de datos que fue introducido en Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., ...& Bloomfield, C. D. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science, 286(5439), 531-537.

El conjunto de datos inicial consistía de 38 muestras de médula ósea obtenidas de pacientes con leucemia aguda en el momento del diagnóstico.
27 con leucemia linfoblástica aguda (ALL) y 11 leucemia myeloide aguda (AML)
Se mide el nivel de expresión de miles de genes utilizando técnicas de *microarrays*.
La matriz de datos consiste de 38 filas y 3571 columnas.

Llamamos

- x_{ij} al nivel de expresión del j-ésimo gen en el i-ésimo paciente.
- $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, $p = 3571$
- $y_i = \begin{cases} 1 & \text{si el } i\text{-ésimo paciente sufre de ALL} \\ 0 & \text{si el } i\text{-ésimo paciente sufre de AML} \end{cases}$

Asumimos que $y_i|\mathbf{x}_i$ sigue una distribución de Bernoulli con :

$$P(y_i = 1|\mathbf{x}_i) = p(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})$$

donde

$$p(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) = \frac{exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}{1 + exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}$$

Llamamos

- x_{ij} al nivel de expresión del j-ésimo gen en el i-ésimo paciente.
- $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$, $p = 3571$
- $y_i = \begin{cases} 1 & \text{si el } i\text{-ésimo paciente sufre de ALL} \\ 0 & \text{si el } i\text{-ésimo paciente sufre de AML} \end{cases}$

Asumimos que $y_i|\mathbf{x}_i$ sigue una distribución de Bernoulli con :

$$P(y_i = 1|\mathbf{x}_i) = p(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})$$

donde

$$p(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) = \frac{exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}{1 + exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}$$

donde $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$, $p = 3571$ y β_i son parámetos desconocidos.

Si tuvieramos muchas más observaciones podríamos considerar usar el método de máxima verosimiltud.

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} L(\beta_0, \beta_1, \ldots, \beta_p)$$

donde

$$L(\beta_0, \beta_1, \ldots, \beta_p) =$$

$$\sum_{i=1}^{n} y_i \log p(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) + (1 - y_i) \log(1 - p(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}))$$

Clasificación:

Si $p(\hat{\beta}_0 + \mathbf{x}_i^T \hat{\boldsymbol{\beta}}) > 0.5$ el paciente i será clasificado como ALL.

Si$p(\hat{\beta}_0 + \mathbf{x}_i^T \hat{\boldsymbol{\beta}}) < 0.5$ el paciente i será clasificado como AML.

Peligros de los datos de alta dimensión

Cuando la cantidad de parámetros a estimar es similar o mayor que la cantidad de observaciones, el método de máxima verosimilitud no se puede aplicar.

Peligros de los datos de alta dimensión

Cuando la cantidad de parámetros a estimar es similar o mayor que la cantidad de observaciones, el método de máxima verosimilitud no se puede aplicar.

El motivo de esto es que haya o no relación entre las covariables y la respuesta se obtendrá un ajuste perfecto $\rightarrow$ Sobreajuste u *Overfitting*

Peligros de los datos de alta dimensión

Cuando la cantidad de parámetros a estimar es similar o mayor que la cantidad de observaciones, el método de máxima verosimilitud no se puede aplicar.

El motivo de esto es que haya o no relación entre las covariables y la respuesta se obtendrá un ajuste perfecto $\rightarrow$ Sobreajuste u *Overfitting*

Es decir, que el modelo ajuste demasiado bien a los datos con los que fue entrenado pero no de buenas pedicciones para nuevas observaciones.

Peligros de los datos de alta dimensión

Cuando la cantidad de parámetros a estimar es similar o mayor que la cantidad de observaciones, el método de máxima verosimilitud no se puede aplicar.

El motivo de esto es que haya o no relación entre las covariables y la respuesta se obtendrá un ajuste perfecto $\rightarrow$ Sobreajuste u *Overfitting*

Es decir, que el modelo ajuste demasiado bien a los datos con los que fue entrenado pero no de buenas pedicciones para nuevas observaciones.

Sobreajuste significa que un modelo menos flexible resultaría en mejores pedicciones para nuevos datos.

Solución: Usar métodos de regularización

Regresión logística regularizada.

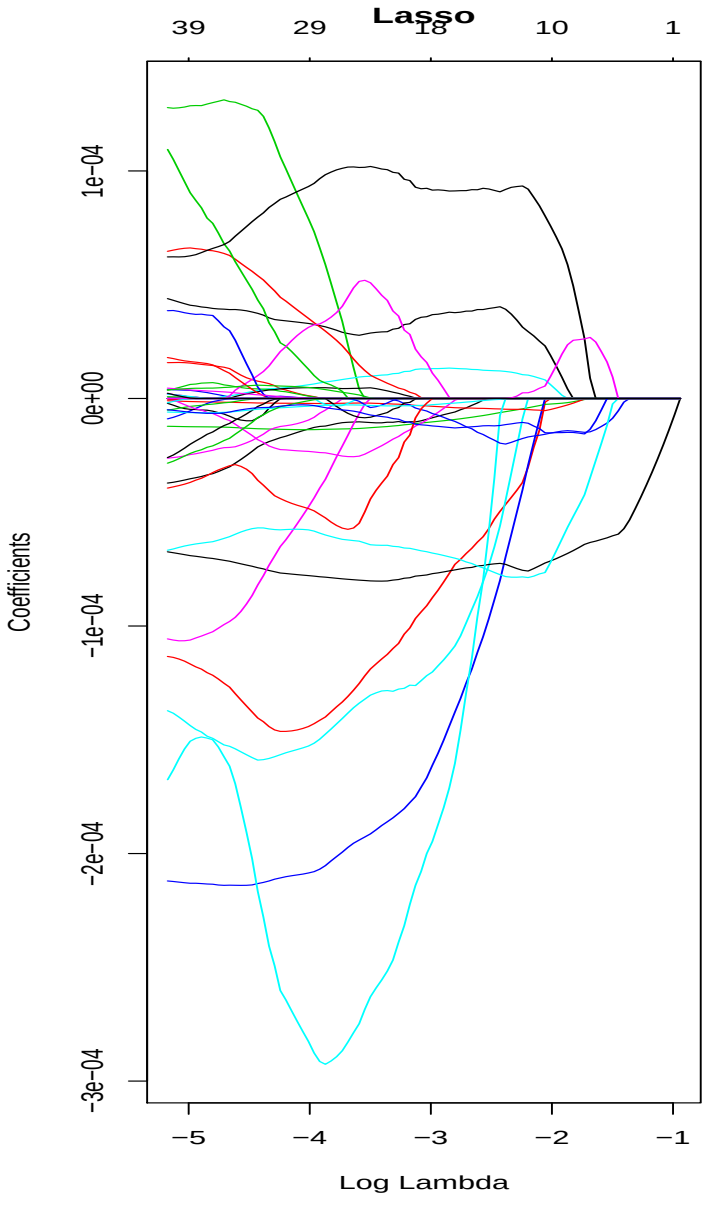
Con penalización Ridge

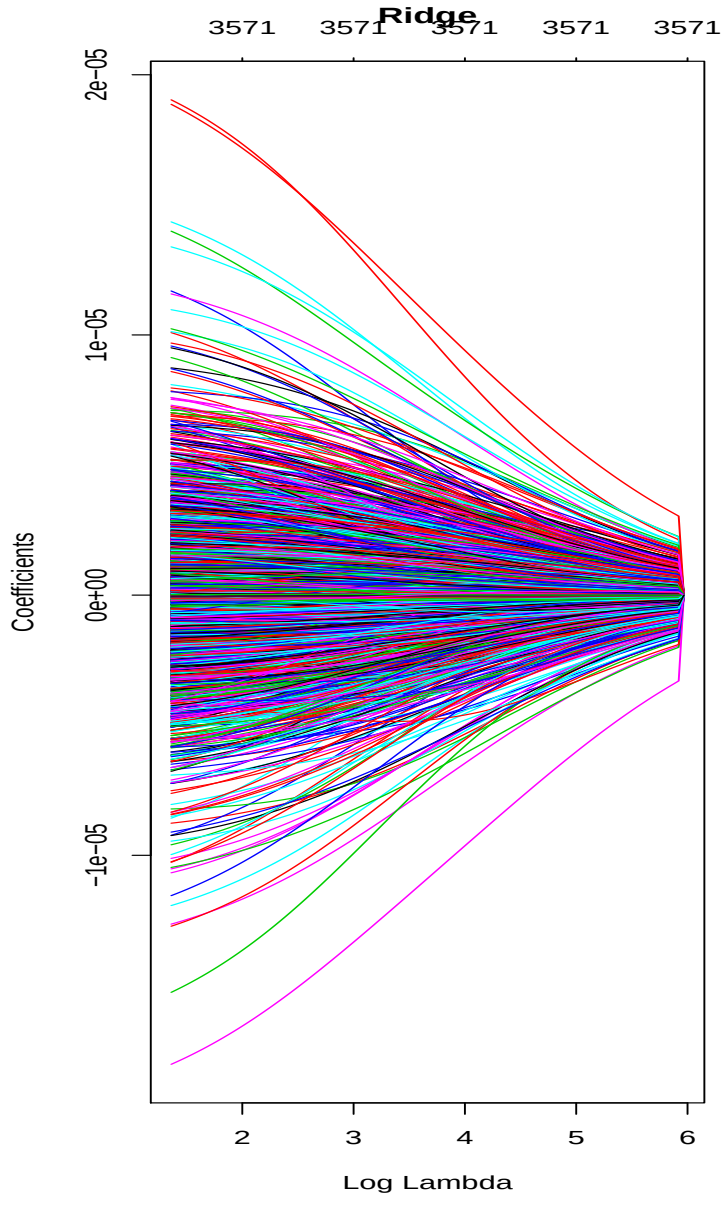
$$\hat{\boldsymbol{\beta}^R} = \arg\max_{\boldsymbol{\beta}} L(\beta_0, \beta_1, \ldots, \beta_p) - \lambda \sum_{j=1}^{p} \beta_j^2$$

Con penalización Lasso

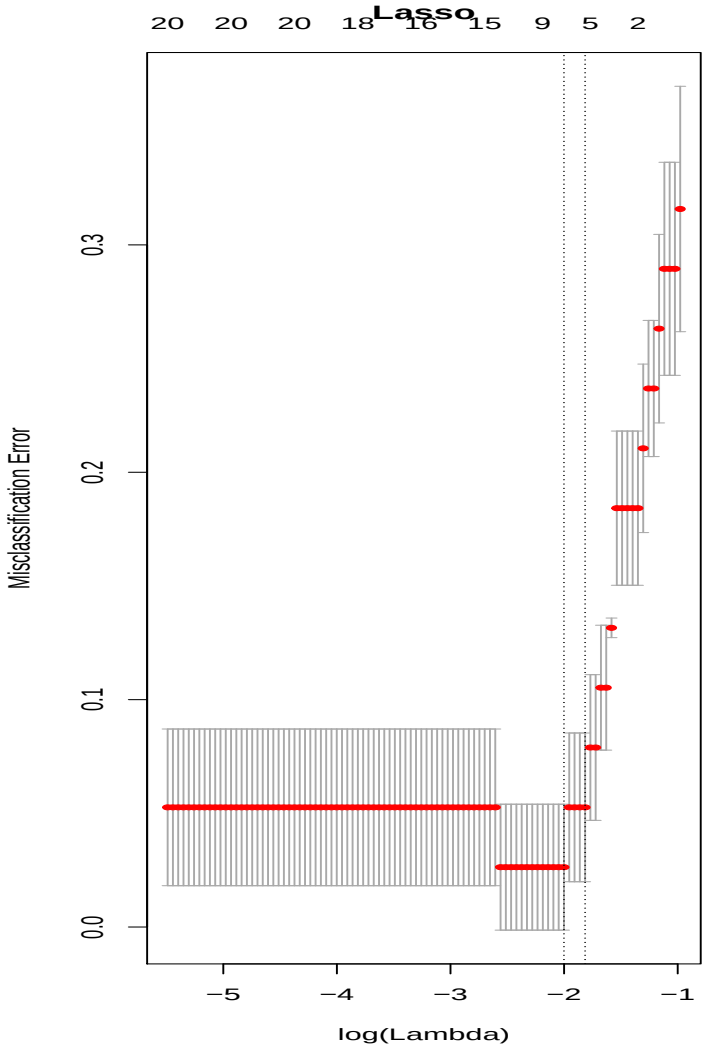
$$\hat{\boldsymbol{\beta}^L} = \arg\max_{\boldsymbol{\beta}} L(\beta_0, \beta_1, \ldots, \beta_p) - \lambda \sum_{j=1}^{p} |\beta_j|$$

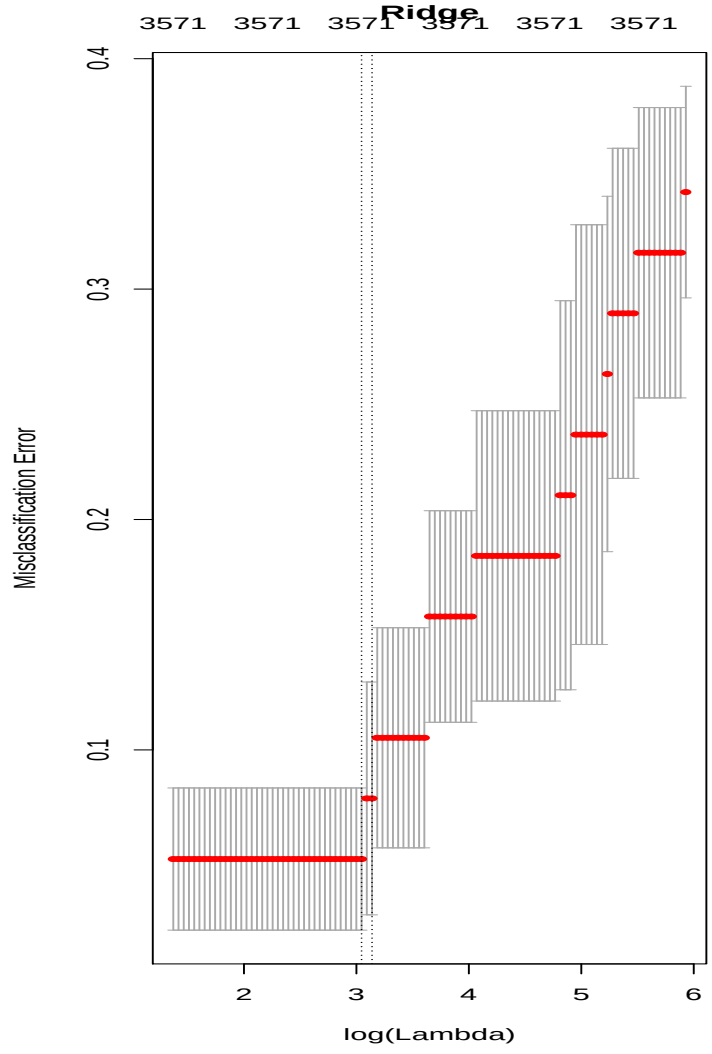
Estimadores ridge y lasso para los datos de leucemia





Error de validación cruzada en función de $\log \lambda$ para datos de leucemia.





Error de validación calculado en una nueva muestra de 34 pacientes

Golub et al disponen luego de una nueva muestra de 34 pacientes. Se los clasifica usando los parámetros estimados usando la muestra de entrenamiento y se compara con la verdadera clasificación. Se obtiene

	Error de clasificación	Cantidad de pacientes mal clasificados
Ridge	0.088	3
Lasso	0.029	1

Comandos de R.

```
library(glmnet)
golub<-read.table("C:/Users/golub.txt,header=TRUE")
ytrain<-golubtrain[,3572]
xtrain<-golubtrain[,1:3571]
fitridge<-glmnet(x,y,family="binomial",alpha=0,standardize=TRUE)
fitlasso<-glmnet(x,y,family="binomial",alpha=1,standardize=TRUE)
par(mfrow=c(1,2))
plot(fitridge);plot(fitlasso)
cv1<-cv.glmnet(x,y,family="binomial",alpha=1)
cv0<-cv.glmnet(x,y,family="binomial",alpha=0)
plot(cv0);plot(cv1)

pred0<-predict(cv0,newx<-xtest,s="lambda.min",type="class")
pred1<-predict(cv1,newx<-xtest,s="lambda.min",type="class")
pred1<-predict(cv1,newx<-xtest,s="lambda.min",type="nonzero")
```

Comandos de R.

```
golubtest<-read.table("C:/Users/golub.txt",header=TRUE)
ytest<-golubtest[,3572]
xtest<-golubtest[,1:3571]

pred1<-predict(cv1,newx<-xtest,s="lambda.min",type="class")
tabla<-table(pred1,ytest)
1-sum(diag(tabla1))/sum(tabla1)

pred0<-predict(cv0,newx<-xtest,s="lambda.min",type="class")
tabla0<-table(pred0,ytest)
1-sum(diag(tabla0))/sum(tabla0)
```

Bibliografía

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112). New York: springer.

Maronna, R. A. R. D., Martin, R. D., & Yohai, V. (2006). Robust statistics. John Wiley & Sons, Chichester. ISBN.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., ...& Bloomfield, C. D. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. science, 286(5439), 531-537.