PRÁCTICA 4 MACHINE LEARNING

Jorge Candia & Alejandro Valbuena

Observación de los datos, preprocesado y elección del modelo

Como esta práctica se está realizando sobre otra ya realizada, resulta sencillo realizar la observación y preprocesado de los datos simplemente volviendo a usar los comandos ya empleados. Por otra parte, la elección de variables será distinta, ya que para comprobar la eficiencia de los distintos tipos de regularización, meteremos todas las variables disponibles en el estudio.

Resumiendo brevemente, en los datos no hay NAs por lo que no hay que eliminar nada, las variables se relacionan de forma muy pobre y todas las discretas vienen ya en factor. Se acabó describiendo la variable logwage en lugar de wage debido a que cambiando la escala a logarítmica, se contrae la dispersión de las muestras, muy alta a escala lineal, y aumenta alrededor de un 15% la R^2, de tan solo 0.3812, pero se baja la RSE del 30% al 6%.

Ya desde el principio separamos los datos en *training* y *test,* eligiendo un ratio del 80% del total de los datos para destinarlos a training.

Presentación del modelo inicial

El modelo base, es el siguiente:

```
base_model <- lm(data=wages, logwage ~
year + age*maritl + race*education + jobclass + health*health_ins)
```

La R^2 resultante es de 0.3812, y el RSME se ha obtenido de la siguiente manera:

```
predictionsTest <- predict(base_model, new=test)
residualsTest  <- test$logwage - predictionsTest

RMSE <- sqrt(mean(residualsTest^2)) #0.2893
PRMSE <- RMSE/mean(test$logwage) * 100 #5.907%
```

Se obtienen las predicciones de test, se restan a los valores reales para obtener los residuos, y a partir de ahí se hace la aritmética necesaria para obtener este estadístico.

Al final, se obtiene que el RMSE puesto en porcentaje es del 5.907%. Este dato será necesario para compararlo con los modelos que usan regularización y así decidir cuál es más adecuado para este estudio. El RMSE de training es del 5.88%, lo cuál sugiere que no existe overfitting en el modelo base, así que se prevé que que emplear regularización no mejore el modelo o que incluso lo empeore.

Lo primero es crear una matriz con los datos de wages, que es lo que se pasará como parámetro al comando glmnet. Este paso es común a los estudios de regularización Ridge y Lasso.

```
#1.- CREO UNA MATRIZ CON LOS DATOS Y UNA LISTA CON LA VARIABLE EXPLICADA
X = model.matrix(logwage ~ year+age*maritl+race*education+jobclass+
health*health_ins, training)[, -1] #El -1 quita el intercept que mete
model.matrix
y = training$logwage

Xtest = model.matrix(logwage ~ year+age*maritl+race*education+jobclass+
health*health_ins, test)[, -1]
ytest = test$logwage
```

CASO 1 -> Estudio de regularización Ridge

La regularización por Ridge busca penalizar variables del modelo con el fin de reducir el overfitting. Se trata de encontrar la beta que minimiza una cierta función para distintos valores del parámetro de regularización lambda. Si lambda es cero, no existe regularización, pero cuanto mayor se hace, mayor es también la penalización de los coeficientes. Una característica importante de este tipo de regularización es que en ningún caso se llegará a hacer cero ninguno de los coeficientes.

Habiendo explicado la teoría, a continuación se explicará el código en R paso por paso:

```
#2.- CREO EL MODELO CON REGULARIZACIÓN RIDGE (alpha = 0)
par(mfrow = c(1, 2)) #plottear las gráficas apilandolas en 2 columnas de 1 fila
fit_ridge = glmnet(X, y, alpha = 0) #alpha = 0 -> Ridge

#2.2.- PLOTS DE PENALIZACIONES DE COEFS PARA DISTITAS LAMBDAS (ninguno es cero)
plot(fit_ridge, label = T)
plot(fit_ridge, xvar = "lambda", label = TRUE)
#Hay tantas variables por los factores y los productos

#3.- CON CV.GLMNET CREO UN MODELO CON CV DE K=10 (DEFAULT) DEL QUE PODEMOS PEDIR
LA LAMBDA ÓPTIMA (MIN Y 1-SE)
par(mfrow = c(1, 1)) #plottear las gráficas apilandolas en 1 columna de 1 fila
(lo normal)
fit_ridge_cv = cv.glmnet(X, y, alpha = 0)
plot(fit_ridge_cv)

fit_ridge_cv$lambda.min #####3.2.- Best Lambda MIN #####
fit_ridge_cv$lambda.1se #####3.3.- Best Lambda with 1-SE #####

#4.- PREDICCIONES CON LOS 2 LAMBDAS
# predict using minimum lambda
predictionsTest <-predict(fit_ridge_cv, Xtest, s = "lambda.min")
```

```
# predict using 1-SE rule lambda, default behavior
predictionsTest <- predict(fit_ridge_cv, Xtest)

#4.2.- ERROR DE LAS PREDICCIONES (elegir arriba si hacerlo con 1se ó min)
residualsTest <- ytest - predictionsTest

summary(residualsTest)
sd(residualsTest)

RMSE <- sqrt(mean(residualsTest^2)) #0.2949 min // 0.3030 1se
PRMSE <- RMSE/mean(test$logwage) * 100 #6.300% min //6.473% 1se
```

Como se ha expresado en el propio código, el estadístico de contraste ha empeorado ligeramente respecto del modelo base, pasando del 5.9% al 6.3%. Cabe mencionar que se probó el desempeño del modelo con 2 lambdas, las descritas en los apartados 3.2 y 3.3.

CASO 2 -> Estudio de regularización Lasso

La regularización Lasso busca también penalizar variables del modelo con el fin de reducir el overfitting. Se trata de nuevo de encontrar la beta que minimiza una cierta función para distintos valores del parámetro de regularización lambda. La principal diferencia con Ridge, además de la función a minimizar, es que aquí sí que los coeficientes pueden llegar a hacerse cero.

De nuevo, se explicará el código paso a paso. Es casi el mismo, lo único que cambia es el parámetro alpha de glmnet:

```
#2.- CREO EL MODELO CON REGULARIZACIÓN LASSO (alpha = 1)
par(mfrow = c(1, 2)) #plottear las gráficas apilandolas en 2 columnas de 1 fila
fit_lasso = glmnet(X, y, alpha = 1) #alpha = 0 -> Lasso

#2.2.- PLOTS DE PENALIZACIONES DE COEFS PARA DISTITAS LAMBDAS
plot(fit_lasso, label = T)
plot(fit_lasso, xvar = "lambda", label = TRUE)

#3.- CON CV.GLMNET CREO UN MODELO CON CV DE K=10 (DEFAULT) DEL QUE PODEMOS PEDIR
LA LAMBDA ÓPTIMA (MIN Y 1-SE)
par(mfrow = c(1, 1))
fit_lasso_cv = cv.glmnet(X, y, alpha = 1)
plot(fit_lasso_cv)

fit_lasso_cv$lambda.min #####3.2.- Best Lambda MIN #####
fit_lasso_cv$lambda.1se #####3.3.- Best Lambda with 1-SE #####

#4.- PREDICCIONES CON LAS 2 LAMBDAS
predictionsTest <-predict(fit_lasso_cv, Xtest, s = "lambda.min")
predictionsTest <- predict(fit_lasso_cv, Xtest) #la 1se está por default
```

```
#4.2.- ERROR DE LAS PREDICCIONES (elegir arriba si hacerlo con 1se ó min)
residualsTest <- ytest - predictionsTest

# RMSE
RMSE <- sqrt(mean(residualsTest^2)) #0.3005 min // 0.2936 1se
# % RMSE:
PRMSE <- RMSE/mean(test$logwage) * 100 #6.419% min //6.272% 1se
```

De nuevo, el estadístico de contraste ha empeorado ligeramente respecto del modelo base, pasando del 5.9% al 6.27%.

CASO 3 -> Estudio de regularización Mixta

La regularización mixta o ElasticNet es una mezcla de la regularización Ridge y Lasso. Una vez más, se trata de minimizar una función, y mediante el parámetro de fusión alpha se pondera el modelo más hacia Ridge (alpha=0) o hacia Lasso (alpha=1), de nuevo para distintos valores de lambda. Para encontrar el mejor modelo, se probarán muchas combinaciones de estos dos parámetros. Se explicará el procedimiento dentro del código:

```
#CREO UN GRID CON LOS VALORES DE ALPHA Y LAMBDA QUE VOY A PROBAR
glmnet_grid <- expand.grid(alpha = c(0,  .1,  .2, .4, .6, .8, 1),
                           lambda = seq(.01, .2, length = 20))

# CV method using 10 folds
glmnet_ctrl <- trainControl(method = "cv", number = 10)
#PRUEBO TODOS LOS MODELOS CON EL COMANDO TRAIN
glmnet_fit <- train(logwage ~
year+age*maritl+race*education+jobclass+health*health_ins, data = training,
                    method = "glmnet",
                    preProcess = c("center", "scale"),
                    tuneGrid = glmnet_grid,
                    trControl = glmnet_ctrl)

glmnet_fit$bestTune #MEJOR RESULTADO: alpha = 0.1, lambda = 0.01, modelo n21

# plotting results
trellis.par.set(caretTheme())
plot(glmnet_fit, scales = list(x = list(log = 2)))

#PREDICCIONES:
#predict es una función genérica de tipo R3 en R que sirve para cualquier objeto
y está preparada para coger siempre la mejor opción del objeto que se pase, pero
aún así se rehará el CV para los alpha y lambda óptimos:
# Rebuilding the model with best lamda value identified
glmnet_grid_best <- expand.grid(alpha = 0.1,lambda = 0.01)
```

```r
# CV method using 10 folds
glmnet_ctrl <- trainControl(method = "cv", number = 10)
glmnet_best <- train(logwage ~
year+age*maritl+race*education+jobclass+health*health_ins, data = training,
                     method = "glmnet",
                     preProcess = c("center", "scale"),
                     tuneGrid = glmnet_grid_best,
                     trControl = glmnet_ctrl)

glmnet_best$results #RESULTADOS DEL MODELO (R^2, RMSE (DE TRAINING), ETC)

#Predicciones con datos nuevos
predictionsTest <- predict(glmnet_best, newdata = test)
residualsTest  <- test$logwage - predictionsTest

summary(residualsTest)
sd(residualsTest)

RMSE <- sqrt(mean(residualsTest^2)) #0.2949 1SE // 0.2949 MIN
PRMSE <- RMSE/mean(test$logwage) * 100  #6.30% 1SE // 6.299% MIN
```

De nuevo, el estadístico de contraste ha empeorado ligeramente respecto del modelo base, pasando del 5.9% al 6.3%.

Criterio de selección del mejor modelo, recomendaciones y conclusión

Como ya se ha ido comentando, al tener el modelo base tan poco overfitting, lo único que hace la regularización es empeorar el estadístico de contraste (RMSE), así que debemos quedarnos con el modelo base sin aplicar ningún tipo de regularización.

De haber tenido overfitting, el RMSE se habría modificado positivamente. El modelo que se debería escoger es aquel que mejore más este parámetro, que naturalmente será el mixto, ya que balancea el alpha en favor de Lasso o de Ridge en función de lo que sea más óptimo, pudiendo llegar a converger en uno u otro si es esto lo mejor. Sin embargo esto no es cierto totalmente porque quizás nuestros intereses pueden ser otros y no tiene sentido usar un mixto. Por ejemplo, un Lasso te va a orientar acerca del mínimo número de variables, cosa que el mixto no lo va a hacer.

En unas pruebas de última hora, hemos ampliado los valores de lambda del grid de la regularización mixta y se observa que el best fit tiene siempre como lambda la más baja posible, lo que evidencia que no es necesario aplicar regularización.