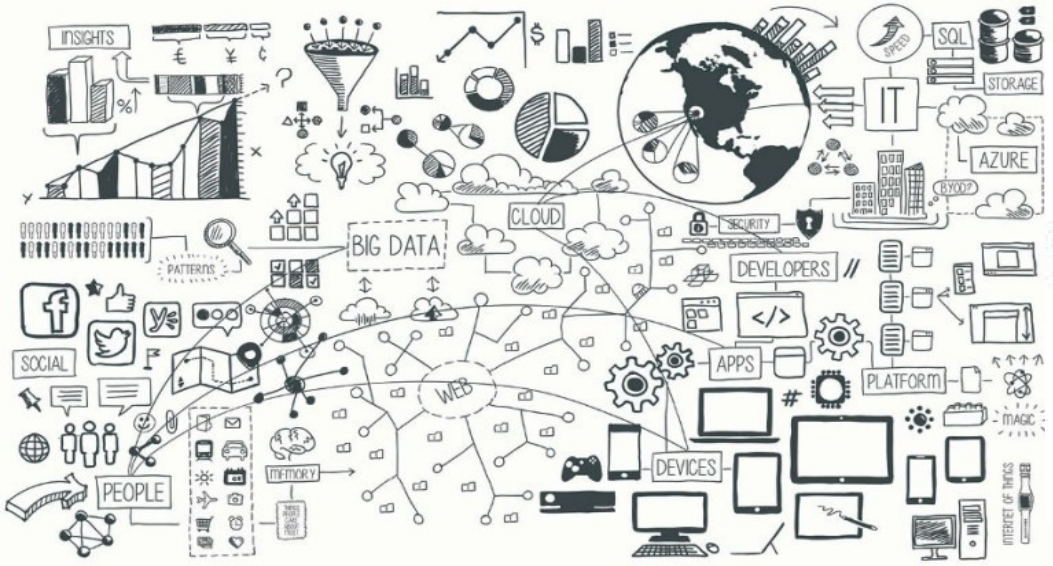
Estadística (M1965)

SELECCIÓN DE VARIABLES Y REGULARIZACIÓN



Jesús Fernández

Grupo de Meteorología

Univ. de Cantabria – CSIC
MACC / IFCA

Estadística

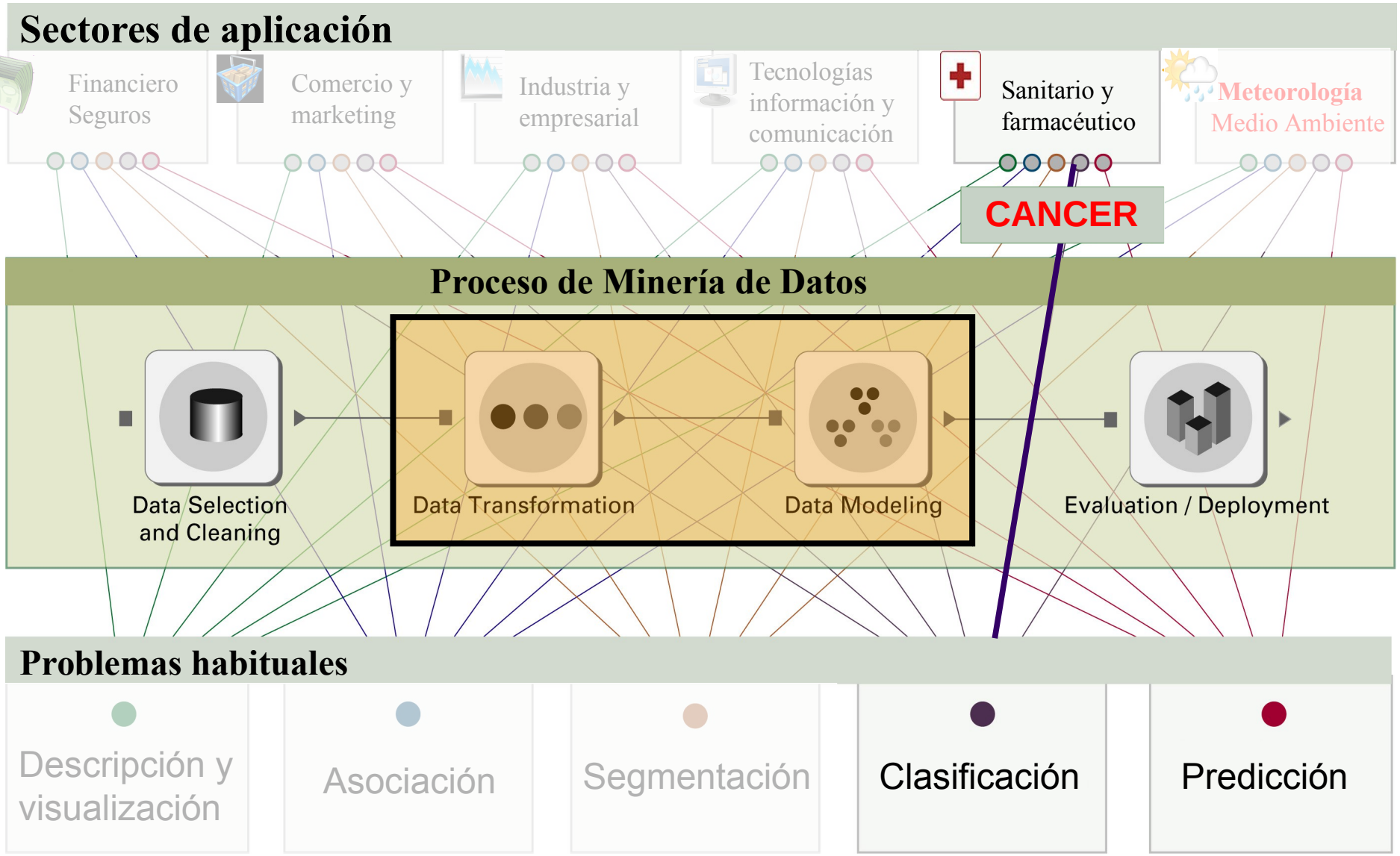
Resumen de estadística
Regresión
Practica: regresión
Regresión logística
Regresión y descenso gradiente
Práctica de clasificación
Estimación de máxima verosimilitud
Bondad de ajuste
Remuestreo
Práctica remuestreo
Selección de variables y regularización
Práctica selección de variables y regularización
Reducción de la dimensión: PCA, LDA
Práctica de PCA y LDA
Examen

Estadística

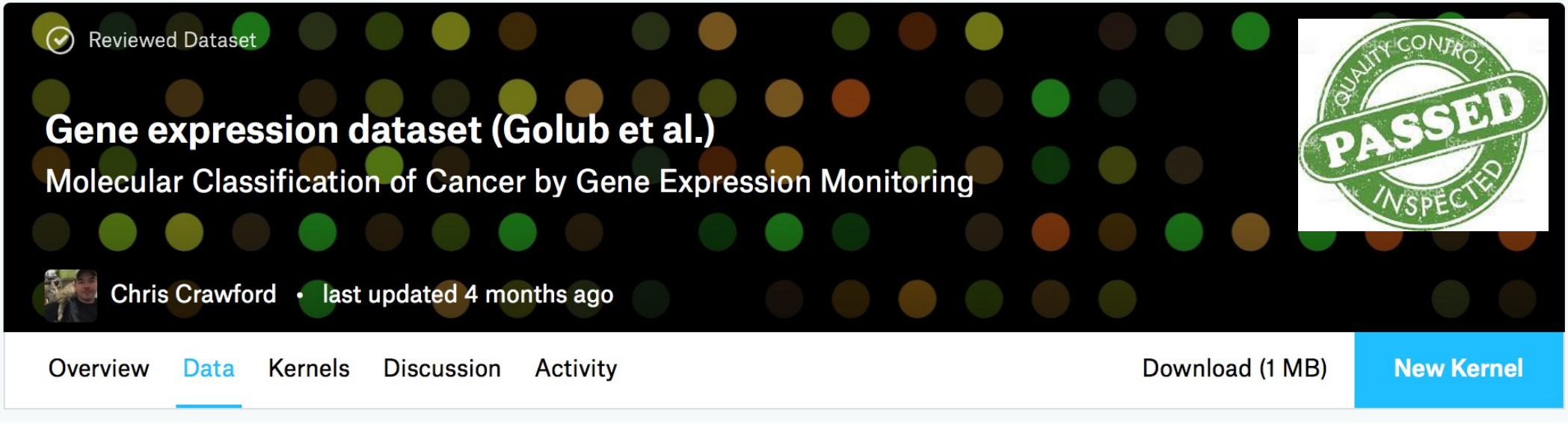
Resumen de estadística
Regresión
Practica: regresión
Regresión logística
Regresión y descenso gradiente
Práctica de clasificación
Estimación de máxima verosimilitud
Bondad de ajuste
Remuestreo
Práctica remuestreo
Selección de variables y regularización
Práctica selección de variables y regularización
Reducción de la dimensión: PCA, LDA
Práctica de PCA y LDA
Examen

Data mining >

Aplazada (sesión de refuezo)
Presentación, introducción y perspectiva histórica
Paradigmas, problemas canonicos y data challenges
Reglas de asociación
Practica: Reglas de asociación
Evaluación, sobrejuste y crossvalidacion
Practica: Crossvalidacion
Arboles de clasificacion y decision
Practica: Arboles de clasificación
T01. Datos discretos
Técnicas de vecinos cercano (k-NN)
Práctica: Vecinos cercanos
Reducción de la dimensión no lineal
Práctica de reducción de la dimensión no lineal
T02. Clasificación
Árboles de decisión: Regresión (CART)
Practica: CART
Ensembles: Bagging and Boosting
Practica Random Forests
T03. Prediccion
Practica Gradient boosting
Técnicas de agrupamiento
Practica: Técnicas de agrupamiento
Practica: El paquete CARET
Examen



Machine learning develop methods for data modelling and prognosis.



Optimization Based Tumor Classification from Microarray Gene Expression Data

Onur Dagliyan[1], Fadime Uney-Yuksektepe[2], I. Halil Kavakli[1], Metin Turkay[3]*

An important use of data obtained from microarray measurements is the classification of tumor types with respect to genes that are either up or down regulated in specific cancer types.

Table 1. Cancer data sets used in this study.

Data set	Samples	Genes	Classes	Reference
Leukemia	72	7129	2	Golub *et al.* (1999)
Prostate cancer	102	12600	2	Singh *et al.* (2002)
Prostate outcome	21	12600	2	Singh *et al.* (2002)

doi:10.1371/journal.pone.0014579.t001

El desequilibrio entre el tamaño muestral y el número de predictores da lugar a los siguientes valores de validación de **train/test**:

Hold Out

```
pred <- predict(modelFit, newdata = trainDF[,-7130])
acc<-confusionMatrix(trainDF$label,pred)
print(acc)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ALL AML
##        ALL  27   0
##        AML   0  11
##
##                Accuracy : 1
##                  95% CI : (0.9075, 1)
##     No Information Rate : 0.7105
##     P-Value [Acc > NIR] : 2.291e-06
##
```

Train → Predicción Perfecta.

```
pred <- predict(modelFit, newdata = testDF[,-7130])
acc<-confusionMatrix(testDF$label,pred)
print(acc)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction ALL AML
##        ALL  10  10
##        AML   7   7
##
##                Accuracy : 0.5
##                  95% CI : (0.3243, 0.6757)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.5679
```

Test → Predicción Aleatoria.

Conociendo los problemas de esta aproximación de validación cruzada,
Es posible corregir este problema usando otra aproximación?

El desequilibrio entre el tamaño muestral y el número de predictores da lugar a los siguientes valores de validación de **train/test**:

Leave-One-Out

```
pred <- predict(modelFit, newdata = trainDF[,-7130])
acc<-confusionMatrix(trainDF$label,pred)
print(acc)
```

El tamaño muestral no permite un **k-fold**, si bien no es esperable obtener resultados mejores a los obtenidos con el **LOOCV**.

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction ALL AML
##        ALL  27   0
##        AML   0  11
##
##                Accuracy : 1
##                  95% CI : (0.9075, 1)
##     No Information Rate : 0.7105
##     P-Value [Acc > NIR] : 2.291e-06
```

Train → Predicción Perfecta.

```
fitControl <- trainControl(method="loocv")
```

Y repetir los pasos anteriores de ajuste y evaluación del modelo en la muestra de test

```
modelFit <- train(label ~ ., data=trainDF, method="glm", trControl=fitControl)
pred <- predict(modelFit, newdata = testDF[,-7130])
acc<-confusionMatrix(testDF$label,pred)
print(acc)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction ALL AML
##        ALL  10  10
##        AML   7   7
##
```

Test → Predicción Aleatoria.

La selección de variables y/o reducción de la dimensión permite construir modelos con **menos grados de libertad** y, por tanto, con menos propensión al **sobreajuste**.

The highest accuracy is obtained with the optimal gene set consisting of 4 genes:

- Myeloperoxidase (M19507-at),
- adipsin (M84526-at),
- CD33 antigen and
- TCF3 transcription factor 3.

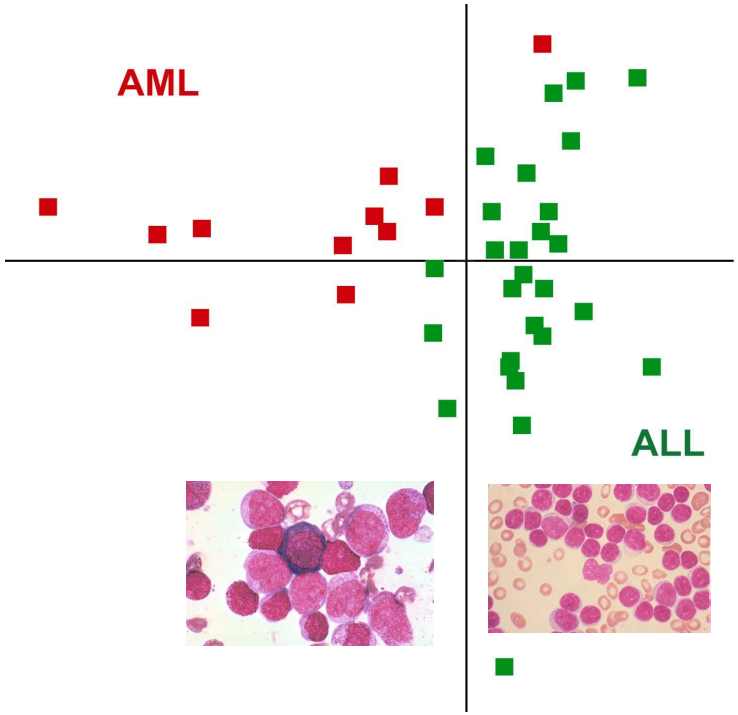


Table 2. Classification results of leukemia (AML-ALL) data set.

Classifier	Test Set	10-CV	LOOCV
HBE	**100**	97.14±0.903	**98.61**
BayesNet	94.12	95.71	95.83
LibSVM	58.82	86.57±10.44	91.67
SMO	97.06	93.14±0.571	94.44
Logistic Regression	91.18	96.86±1.67	**98.61**
RBF Network	97.06	**97.43±1.07**	97.22
IBk	97.06	96.00±1.40	95.83
J48	94.12	89.14±1.94	90.28
Random Forest	94.12	93.14±1.07	90.2

doi:10.1371/journal.pone.0014579.t002

Propiedades básicas de un conjunto de variables predictoras

Proximidad

Es el grado en el que un conjunto de variables predictoras es capaz de explicar la variable a predecir (predictando). Variables próximas dan lugar a predicciones más robustas.

Multicolinealidad

Alta correlación entre dos o más variables predictoras. Puede afectar negativamente la capacidad predictiva del modelo.

Dimensionalidad

Número de predictores. Un número alto de predictores puede dar lugar a modelos sobre-ajustados con menor capacidad de generalización.

Propiedades básicas de un conjunto de variables predictoras

¿Cómo tratar con la …

Proximidad **… irrelevancia y …**

Es el grado en el que un conjunto de variables predictoras es capaz de explicar la variable a predecir (predictando). Variables próximas dan lugar a predicciones más robustas.

Multicolinealidad **… la redundancia?**

Alta correlación entre dos o más variables predictoras. Puede afectar negativamente la capacidad predictiva del modelo.

Dimensionalidad

Número de predictores. Un número alto de predictores puede dar lugar a modelos sobre-ajustados con menor capacidad de generalización.

Aproximaciones al problema de la preparación de un conjunto "óptimo" de predictores

Reducción de la dimensión
Selección de las primeras componentes principales. Se encuentra un conjunto reducido de nuevas variables predictoras que explican gran parte de la variabilidad original del conjunto completo.

+ Menos variables; se evita el sobreajuste

+ Variables ortogonales, eliminan el problema de la colinealidad

– Mezcla de variables: pérdida de interpretabilidad de los resultados

Próxima semana

Aproximaciones al problema de la preparación de un conjunto "óptimo" de predictores

Reducción de la dimensión
Selección de las primeras componentes principales. Se encuentra un conjunto reducido de nuevas variables predictoras que explican gran parte de la variabilidad original del conjunto completo.

Selección de variables
se identifica un subconjunto de variables predictoras las cuales son usadas para construir el modelo, desechando el resto.

Regularización
En este caso, aunque se consideran todas las variables predictoras, se modifica el modelo de modo que parte de los parámetros se hacen nulos, o casi nulos. Dependiendo del grado de regularización se puede conseguir que algunos coeficientes se anulen, dando lugar a una selección de variables.

Selección de variables

– **Best subset selection:**

+ Considera todas las combinaciones posibles (2^p)
+ Coste computacional alto
+ Adecuado para un conjunto reducido de variables

– **Stepwise:**

+ Forward → Modelo inicial: sin predictores (M = 0)
+ Backward → Modelo inicial: todos los predictores (M = p)

(1) Se parte del modelo inicial

(2) Se consideran todos los modelos posibles incrementando/decrementando un predictor (M = 1 y M = p – 1).

(3) Se considera el mejor de todos ellos y se vuelve al paso (2).

(4) Una vez tenemos un modelo para cada M=0,..., p, seleccionamos el mejor de ellos.

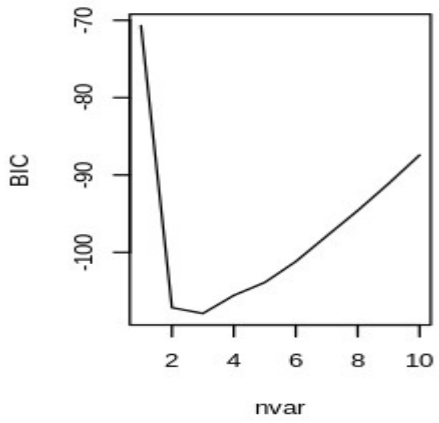
Para hacer selección de variables en modelos lineales en R: `library(leaps)`

La función `regsubsets` permite tanto la selección del mejor conjunto (`method="exhaustive"` por defecto), como las aproximaciones hacia adelante (`method="forward"`) y hacia atrás (`method="backward"`):

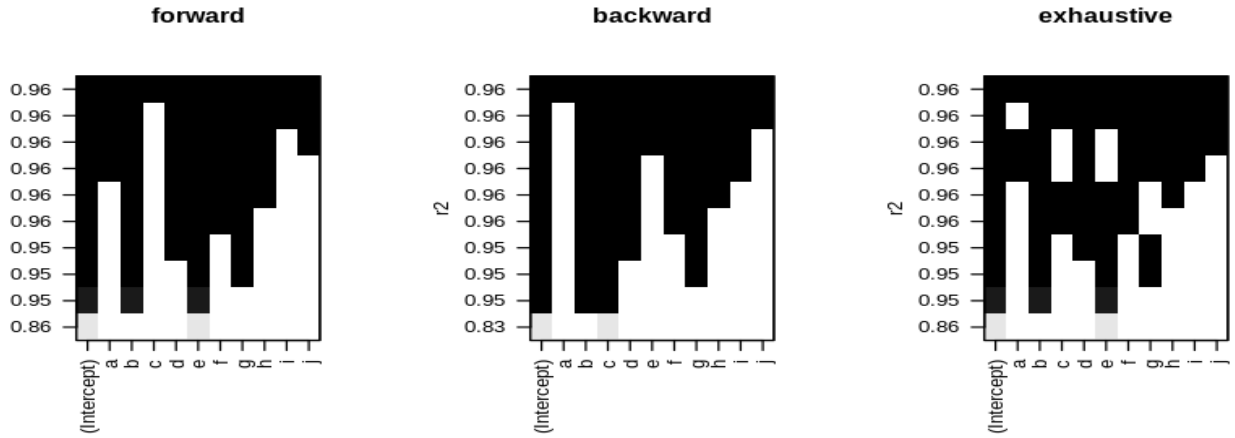
```
reg.fit <- regsubsets(y~., dataframe, nvmax = ..., method = ...)
reg.summary <- summary(reg.fit)
```

El valor de nvmax determina el número de variables máximo a considerar. Por defecto son sólo 8, así que convendrá ajustarlo.

`plot(reg.summary$bic)`



`plot(reg.fit, scale="r2")`



Otras funciones y librerías: `step, bestglm, MASS::stepAIC`

Aproximaciones al problema de la preparación de un conjunto "óptimo" de predictores

Reducción de la dimensión
Selección de las primeras componentes principales. Se encuentra un conjunto reducido de nuevas variables predictoras que explican gran parte de la variabilidad original del conjunto completo.

Selección de variables
se identifica un subconjunto de variables predictoras las cuales son usadas para construir el modelo, desechando el resto.

Regularización
En este caso, aunque se consideran todas las variables predictoras, se modifica el modelo de modo que parte de los parámetros se hacen nulos, o casi nulos. Dependiendo del grado de regularización se puede conseguir que algunos coeficientes se anulen, dando lugar a una selección de variables.

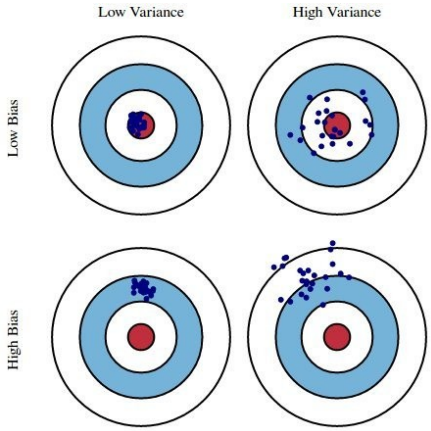
Regularization

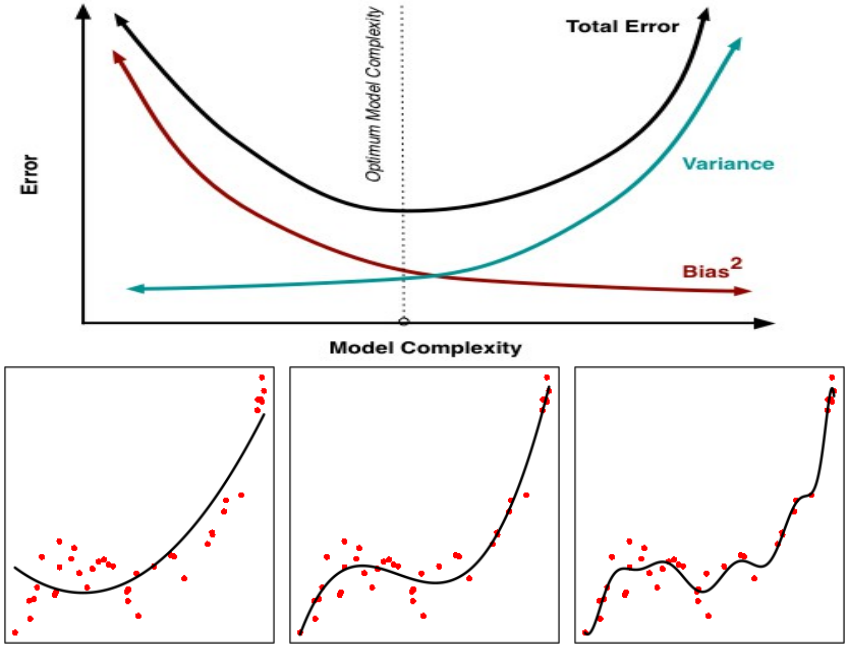
$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \text{Regularization term}$$

Regularization introduces a **penalty term in the cost function** to be minimized in order to prevent overfitting. The estimated parameters are biased due to this extra term, but the variance of the estimated parameter may decrease due to decreases in complexity and redundancy in the model.

Why does it avoid overfitting?

- Limits the norm of the weights
- Bias/variance trade-off





Ideally, one wants to choose a model that both accurately captures the regularities in its training data, but also generalizes well to unseen data.

L2 Regularization or Ridge Regression

Objective function:

$$PRSS(\beta)_2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \|\beta\|_2^2$$

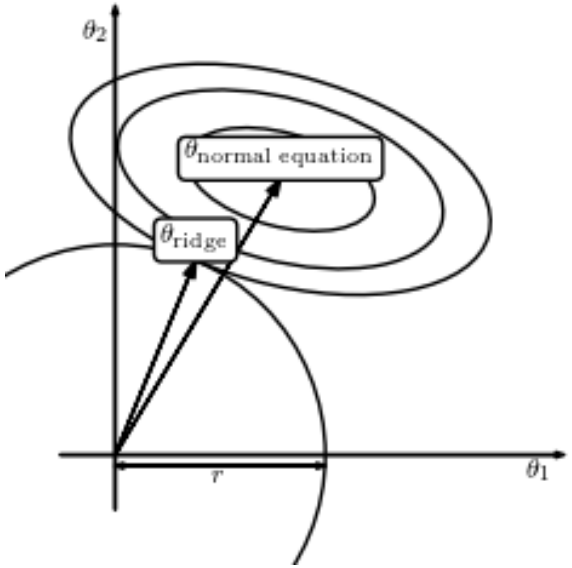
Properties:

1) Ridge regression seeks coefficient estimates that fit the data well, by making the RSS small
2) However, the second term, called a shrinkage penalty, has the effect of shrinking the estimates of β towards zero.
3) The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates.
4) Selecting a good value for is λ critical; cross-validation is used for this.

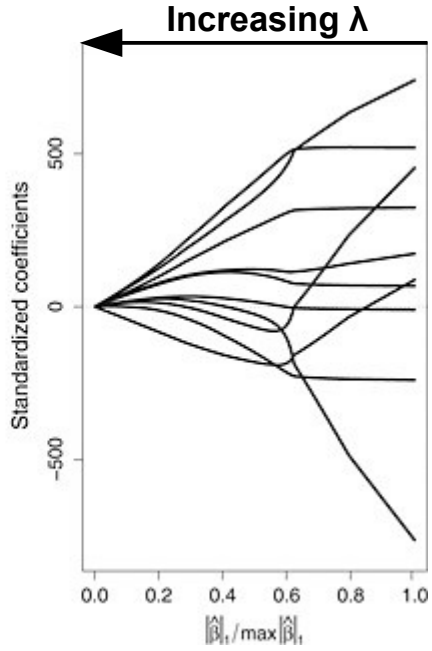
Deriving the above penalized RSS w.r.t. β and equating to zero, we obtain the normal equation set:

$$(X^t X + \lambda I)\beta = X^t y$$

Which leads to a non-singular linear system of equations even if collinearities existed in the design matrix X.



(c) Jake VanderPlas (astroml.org)



(c) Nick Ryan (nickcdryan.com)

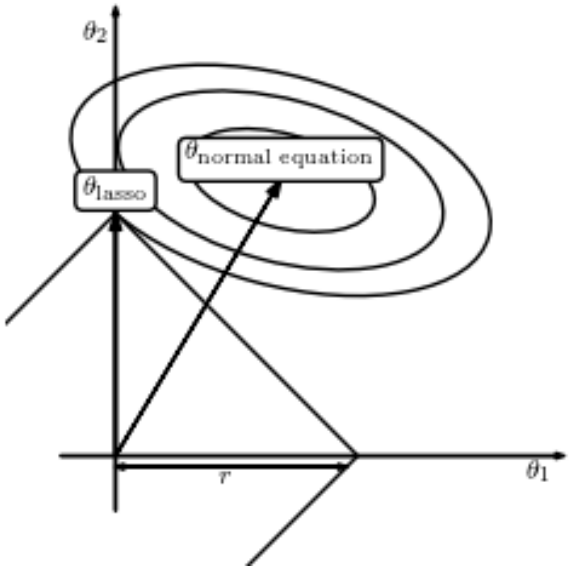
L1 Regularization or LASSO Regression

Objective function:

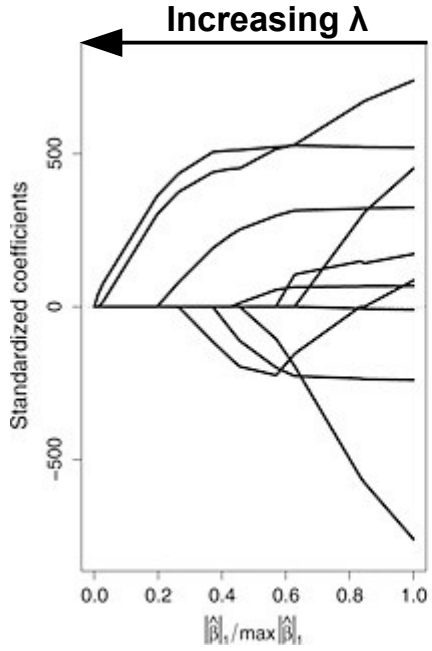
$$PRSS(\beta)_1 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \|\beta\|_1$$

Properties:

1) However, in the case of LASSO*, the penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
2) Hence, LASSO performs variable selection and thus, more interpretable models.
3) The tuning parameter serves to control the relative impact of these two terms on the regression coefficient estimates.
4) Again, selecting a good value for λ is critical; cross-validation is used for this.



(c) Jake VanderPlas (astroml.org)



(c) Nick Ryan (nickcdryan.com)

* Least absolute shrinkage and selection operator

Para hacer regularización de variables para un GLM en R: `library(glmnet)`

El λ óptimo se puede obtener mediante validación cruzada, utilizando la función

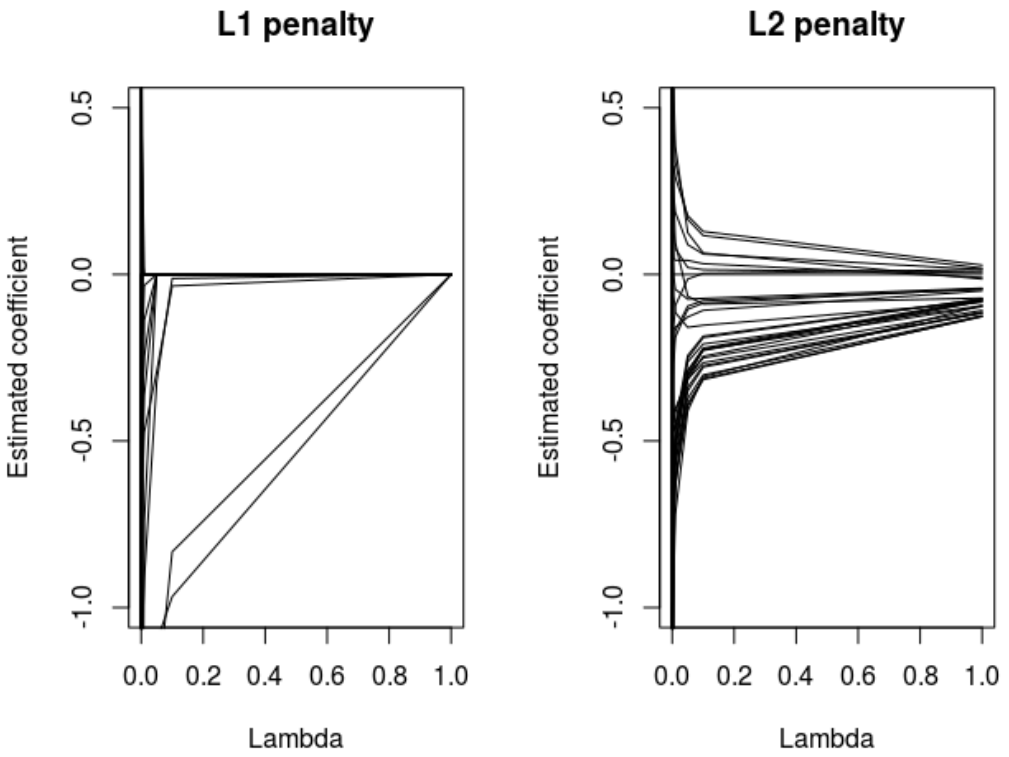
```
cv.glmnet(x, y, alpha = 0, ...)
```

El valor `alpha = 0` representa la regularización L2. Para LASSO (L1) usaríamos `alpha = 1`. Esta función devuelve una lista, entre cuyos valores está `lambda.min`, el valor de lambda que minimiza el error en la validación cruzada (por defecto, un 10-fold).

Una vez decidido el valor de λ a usar, podemos construir el modelo con:

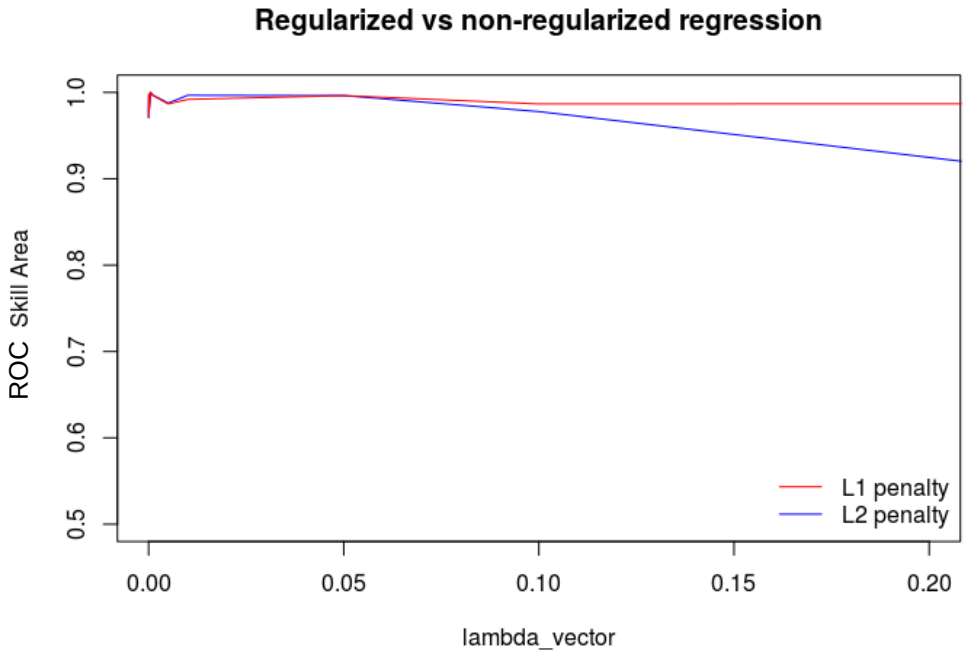
```
L1model <- glmnet(x, y, alpha = 1, lambda = valor, ...)
```

```
library(glmnet)
lambda_vector <- c(1,0.1,0.05,0.01,0.005,0.001,0.0005,0.0001,0.00005,0.00001,0)
coef_l1 <- sapply(1:length(lambda_vector), function(z){
  as.matrix(coef(glmnet(
    x = as.matrix(df[,-1]), y = as.matrix(df[,1]), alpha = 1, lambda = lambda_vector[z], family = "binomial"
))))})
coef_l2 <- sapply(1:length(lambda_vector), function(z){
  as.matrix(coef(glmnet(
    x = as.matrix(df[,-1]), y = as.matrix(df[,1]), alpha = 0, lambda = lambda_vector[z], family = "binomial"
))))})
par(mfrow = c(1,2))
plot(lambda_vector,coef_l1[2,], type = "l", ylim = c(-1,0.5),
  xlab = "Lambda", ylab = "Estimated coefficient", main = "L1 penalty"
)
for (i in 3:nrow(coef_l1)) {
  lines(lambda_vector,coef_l1[i,], type = "l")
}
plot(lambda_vector,coef_l1[2,], type = "l", ylim = c(-1,0.5),
  xlab = "Lambda", ylab = "Estimated coefficient", main = "L2 penalty"
)
for (i in 3:nrow(coef_l1)) {
  lines(lambda_vector,coef_l2[i,], type = "l")
}
```



```
library(verification)
e <- sapply(1:length(lambda_vector), function(z){
  ind <- sample(x = 1:number_of_examples,size = round(number_of_examples*0.80),replace = FALSE)
  ind2 <- setdiff(1:number_of_examples,ind)
  cf2 <- glmnet(x = as.matrix(df[ind,-1]), y = as.matrix(df[ind,1]), alpha = 1, lambda = lambda_vector[z], family = "binomial")
  cf3 <- glmnet(x = as.matrix(df[ind,-1]), y = as.matrix(df[ind,1]), alpha = 0, lambda = lambda_vector[z], family = "binomial")
  p2 <- predict.glmnet(cf2,as.matrix(df[ind2,-1]),type = "response")
  p3 <- predict.glmnet(cf3,as.matrix(df[ind2,-1]),type = "response")
  e2 <- roc.area(df[["diagnosis"]][ind2],p2)$A
  e3 <- roc.area(df[["diagnosis"]][ind2],p3)$A
  c(e2,e3)
})

plot(lambda_vector,e[1,], xlim = c(0,0.05), type = "l",
  xlab = "lambda_vector", ylab = "ROC Skill Area", main = "Regularized vs non-regularized regression", col = "blue"
)
lines(lambda_vector,e[2,], col = "red")
legend("bottomright","groups", legend = c("L1 penalty", "L2 penalty"), lty = 1,  bty = "n",  col = c("red","blue"), cex = 1)
```



There is a gain with a regularized version but not very significant because this example does not have a lot of explanatory variables…

Regularized versions are very relevant when the number of explanatory variables is higher than the number of observations:

→ **GENE EXPRESSION DATASET**