Práctica 3

Regresión Logística

Javier Herrer Torres (NIP: 776609)

Aprendizaje automático
Grado en Ingeniería Informática

Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza

Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza
Curso 2020/2021

1. Objetivo

El objetivo es aplicar regresión logística en casos sencillos de clasificación binaria, utilizando técnicas de regularización y validación cruzada.

2. Estudio previo: K-fold Cross-Validation

Para la implementación se ha seguido el algoritmo de la transparencia 13 del tema de Regularización.

```
function [hypothesis, best_model] = kfold_cross_validation(k, X, y)
best_model = 0;
best_errV = Inf;
errT_aux = 0;
% para los distintos valores de los hyper-parámetros
for model = logspace(-10,0)
    err_T = 0;
    err_V = 0;
    %separar N/k ejemplos para validación
    for fold = 1:k
        [Xcv, ycv, Xtr, ytr] = particion(fold, k, X, y);
        % aprender con el resto
        h = minFunc(@CosteLogReg, theta_ini, options, ...
            Xtr, ytr, lambda);
        err_T = err_T + tasa_error(h, Xtr, ytr);
        err_V = err_V + tasa_error(h, Xcv, ycv);
    end
    % calcular el error medio de las k veces
    err_T = err_T / k;
    err_V = err_V / k;
    if  (err_V < best_errV)
        % guarda el mejor valor de los hyper-parámetros
        best_model = model;
        best_errV = err_V;
    end
    errT_aux = err_T;
end
% aprender de nuevo con todos
hypothesis = regresion(X, y, best_model);
```

3. Regresión logística básica

Se ha predicho qué alumnos serán admitidos a una universidad, en función de la calificación obtenida en dos exámenes, aprendiendo a partir de los datos del fichero `exam_data.txt`. Se ha separado un 20 % de los datos para test mediante

la función `particion.m`. Se ha empleado la función de coste `CosteLogReg.m` que aparece en las transparencias para resolver la regresión logística mediante optimización avanzada con `minFunc`. Se ha usado un $\lambda = 0$ al no introducir aún regularización. Se han obtenido los siguientes resultados para la tasa de errores con los datos de entrenamiento y con los datos de test:

$$E_{train}(\theta) = 0{,}1125$$

$$E_{test}(\theta) = 0{,}1000$$

Para un alumno que ha sacado 45 puntos en el primer examen, se ha dibujado a gráfica de la figura 1 con la probabilidad de ser admitido en función de la calificación del segundo examen. En ella se puede observar que, un alumno **podría ser admitido con una nota superior a 80**.

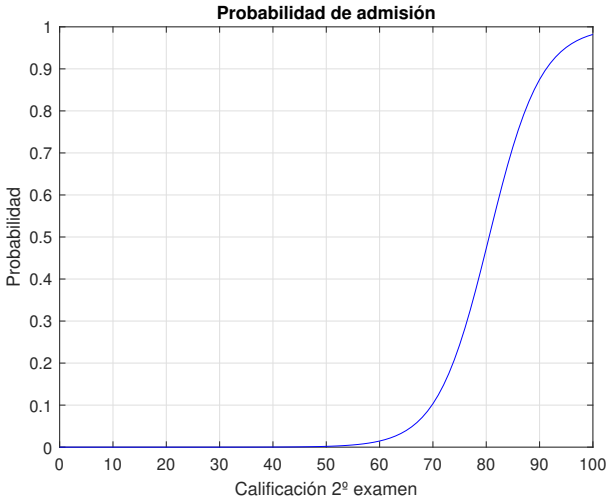

Figura 1: Probabilidad de admisión en función del segundo examen.

4. Regularización

Se ha predicho qué microchips serán aceptados o rechazados, en función de los resultados de dos tests, aprendiendo a partir de los datos del fichero `mchip_data.txt`. Se ha separado un 20 % de los datos para test mediante la función `particion.m`. Se ha empleado regresión logística regularizada con expansión de funciones base mediante la función `mapFeature.m` proporcionada. Se ha elegido el parámetro de regularización λ mediante k-fold cross-validation, obteniendo $\lambda = 1{,}3257 \cdot 10^{-4}$. En la figura 2 se encuentran dibujadas las curvas de evolución de las tasas de errores con los datos de entrenamiento y de validación.

Es importante destacar que, para facilitar el estudio de los resultados se ha eliminado la aleatoriedad de la permutación inicial de los datos.

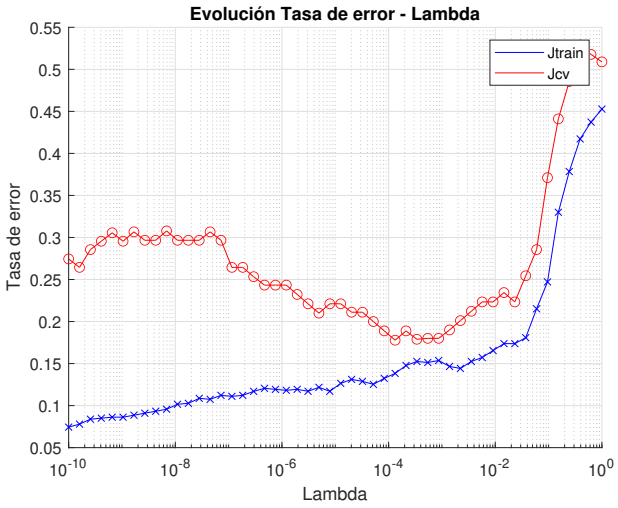

Figura 2: Curvas de evolución de las tasas de errores

Finalmente, se ha entrenado con todos los datos (excepto los de test) el mejor modelo encontrado y el modelo con $\lambda = 0$, obteniendo las siguientes tasas de error:

$$E_{test}(\lambda = 1{,}3257 \cdot 10^{-4}) = 0{,}1250$$

$$E_{test}(\lambda = 0) = 0{,}2083$$

En las figuras 3 y 4 están dibujadas las correspondientes superficies de separación.

5. Precisión/Recall

Con el mejor modelo obtenido, se han utilizado los **datos de test** para calcular la matriz de confusión de la figura 7, y los siguientes valores de precisión y recall:

$$Precision = 0{,}95$$

$$Recobrado = 0{,}86$$

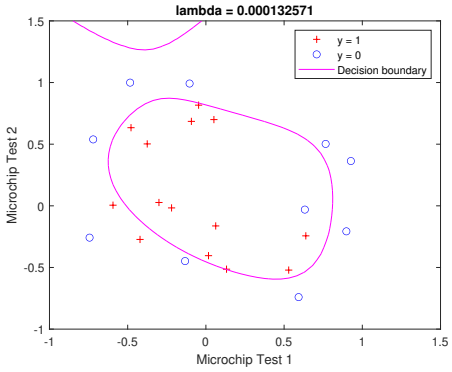

Figura 3: Superficie de separación del mejor modelo encontrado

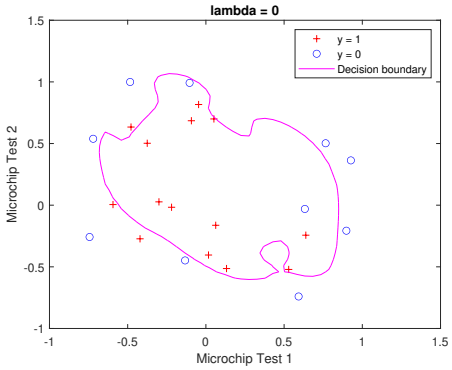

Figura 4: Superficie de separación del modelo con $\lambda = 0$

5.1. Compromiso entre Precision y Recall

Si se quiere que el 95 % de los chips aceptados sean buenos, se debería aumentar el umbral, para aumentar la precisión, aunque disminuiría el recobrado. Este razonamiento se puede inferir de la figura 6. Además, para arrojar una cifra concreta del umbral se ha empleado la curva de precision y recall de la figura 7 con los datos de entrenamiento ya que los de test son insuficientes para este propósito. El umbral con el que se obtendría una precisión del 95 % es:

$$umbral = 0{,}861$$

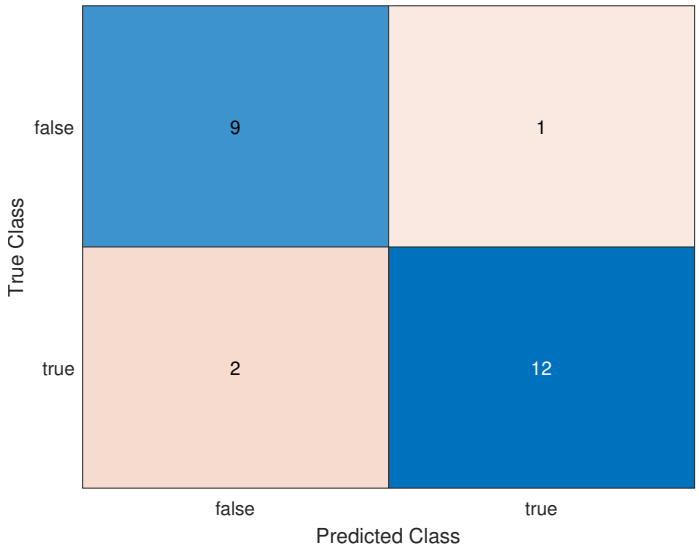

Figura 5: Matriz de confusión

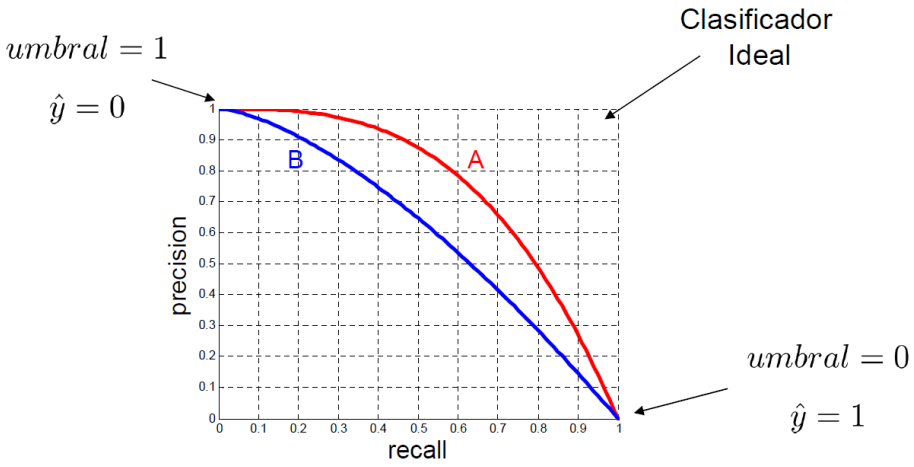

Figura 6: Compromiso entre Precision y Recall

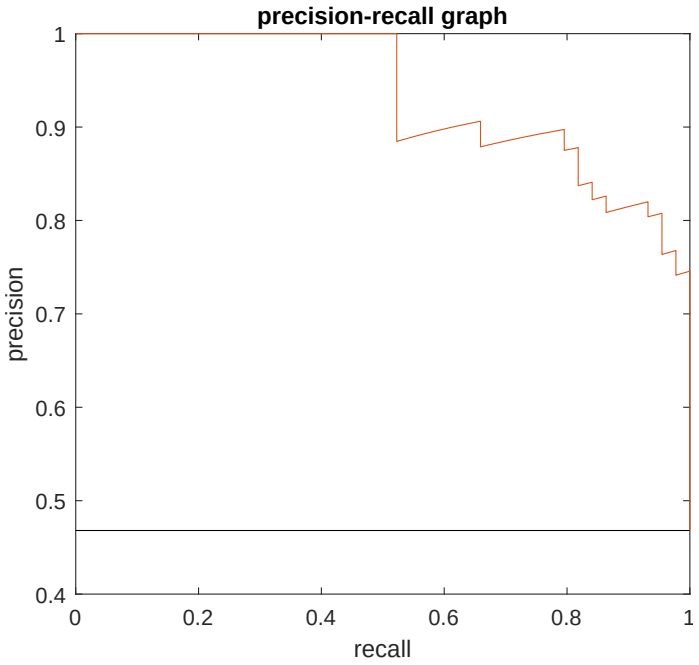

Figura 7: Curva de precision-recall