Práctica 4

Regresión Logística Multi-Clase

Javier Herrer Torres (NIP: 776609)

Aprendizaje automático
Grado en Ingeniería Informática

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

1. Objetivo

El objetivo es resolver mediante regresión logística un problema real de clasificación multi-clase: el reconocimiento de dígitos manuscritos. Utilizaremos una versión reducida del conjunto de datos MNIST. La figura muestra un ejemplo de los mismos. Cada muestra es una imagen de 20x20 píxeles. Como atributos para la clasificación utilizaremos directamente los niveles de intensidad de los 400 píxeles.

2. Estudio previo: Algoritmo de entrenamiento

Se ha escrito el algoritmo de entrenamiento y clasificación multi-clase utilizando regresión logística regularizada. El paquete `minFunc` se ha usado con la opción «**pnewton0**»: *Preconditioned Hessian-Free Newton*, ya que la mejor opción, «newton», es demasiado costosa en el cálculo del Hessiano a la hora de invertir la matriz de D=400.

```
function [h_total, err_T, err_V] =
                    multiclass_training(lambda, Xcv, ycv, Xtr, ytr)
[~, col] = size(Xtr);
theta_ini = zeros(col,1);
options.display = 'iter';
options.method = 'pnewton0';
for clase = 1:10
    ytrain = (ytr == clase);
    h = minFunc(@CosteLogReg, theta_ini, options, ...
        Xtr, ytrain, lambda);
    h_total(:, clase) = h;
end
err_T = tasa_error(h_total, Xtr, ytr);
err_V = tasa_error(h_total, Xcv, ycv);
end

function error = tasa_error(theta, X, y)
% Predicción de la salida
h = 1./(1+exp(-(X*theta)));
%Máximos de cada fila
[~, y_pred] = max(h, [], 2);
% Tasa de error
error = mean(y_pred ~= y);
end
```

Se ha entrenado para todas las clases, y una vez que se tienen los thetas para todas las clases, se ha calculado la salida de la sigmoide para cada clase y calculado el máximo. Con ese valor que se encuentra es con el que se obtienen los errores de training y validación. La salida predicha se obtiene con el índice de la columna que alberga el **máximo de cada fila**:

$$\hat{y}^{(i)} = argmax_j h_\theta^{(j)}(x^{(i)})$$

3. Regresión logística regularizada

Se ha programado el entrenamiento y clasificación multi-clase basándose en el código de la práctica anterior. A diferencia de prácticas anteriores, no se han expandido los atributos debido al gran número de atributos (400).

Se ha separado un 20 % de los datos para validación ($k = 5$), y se ha resuelto la regresión logística regularizada. En la figura 1 se encuentra dibujada la gráfica de tasa de errores en función del parámetro landa. Se puede observar sub-ajuste a la derecha, y sobre-ajuste a la izquierda del punto ideal que se encuentra en:

$$best\_\lambda = 6{,}9519 \cdot 10^{-4}$$

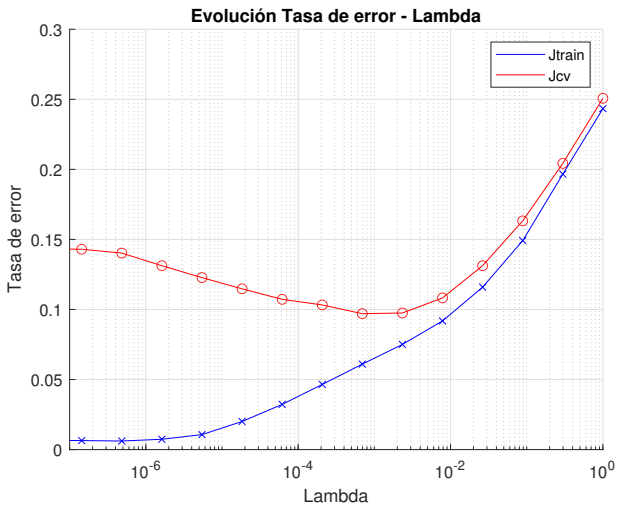

Figura 1: Gráfica de tasa de errores en función del parámetro landa

Posteriormente, se ha entrenado con ese mejor valor de λ y se han obtenido las siguientes tasas de error:

$$E_{train}(\theta) = 0{,}0645$$

$$E_{test}(\theta) = 0{,}1020$$

4. Matriz de confusión y Precision/Recall

Se ha re-entrenado con todos los datos para el mejor valor de λ, y se han utilizado los datos de test para calcular la matriz de confusión de la figura 2 junto con los valores de precisión y recall para cada dígito, y los globales. En la figura 3 se pueden apreciar dígitos problemáticos:

- El dígito 3 se confunde con el 2 y con el 5.
- El dígito 5 se confunde con el 3.
- El dígito 8 se confunde con el 1 y con el 5.

De los que damos como positivos, el 89,82 % **lo son realmente. De los casos positivos, detectamos el** 89,80 %**.**

Dígito Predicho

Dígito Real	1	2	3	4	5	6	7	8	9	0	Suma	Recall
1	97	1						2			100	0,97
2	2	85	3	1	1	2	2	3		1	100	0,85
3	1	5	86		5		1	2			100	0,86
4	1	3		89		2		2	3		100	0,89
5	1		6	2	86	2		3			100	0,86
6						98		1	1		100	0,98
7	2	2		2			92		2		100	0,92
8	4	1	2	2	4	2		83	2		100	0,83
9	1		3	3	1		4	2	86		100	0,86
0				1	2	1				96	100	0,96
Suma	109	97	100	100	99	107	99	98	94	97	1000	0,8980
Precision	0,89	0,88	0,86	0,89	0,87	0,92	0,93	0,85	0,91	0,99	0,8982	

Figura 2: Matriz de confusión

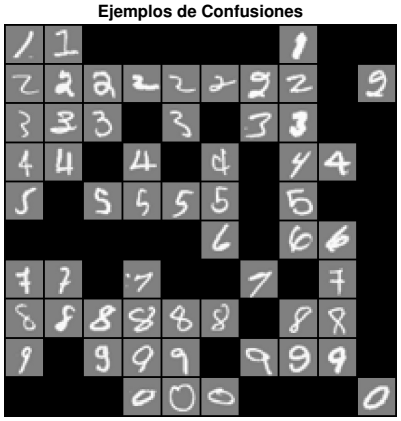

Figura 3: Confusiones

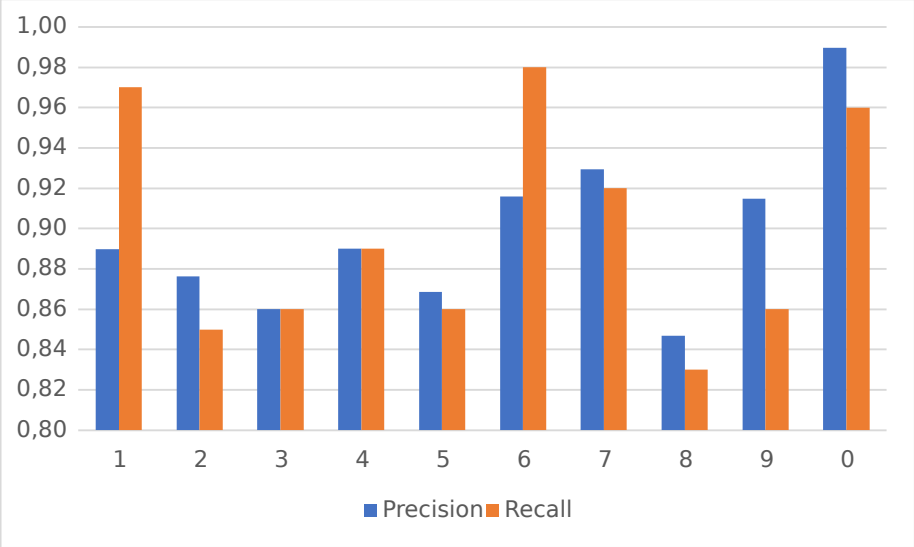

Figura 4: Valores de precisión y recall para cada dígito