Aprendizaje automático

Práctica 4. Regresión Logística Multi-Clase

Universidad de Zaragoza, curso 2022/2023

Juan Eizaguerri Serrano 816079

Clasificación multiclase

Para resolver el problema de clasificar imágenes de números se va a utilizar la regresión logística multi-clase one-vs-all en la que se entrena un clasificador para cada clase posible de salida y la clase predicha para una entrada es la que tiene más probabilidad.

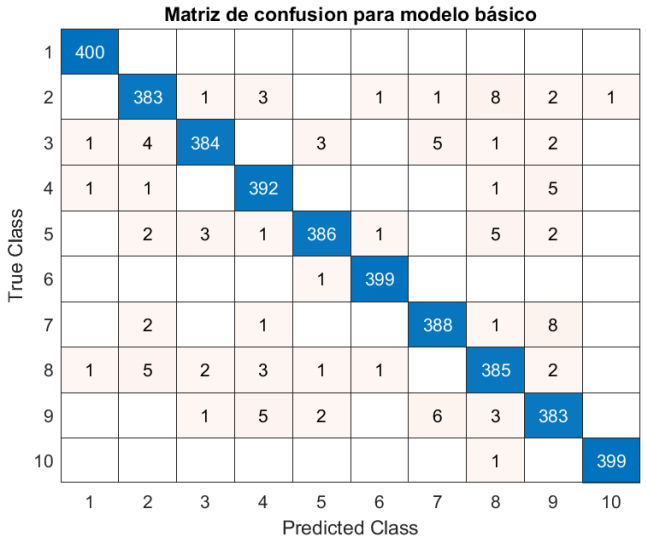
Contamos con el dataset MNIST, que ofrece 5000 imágenes 400x400 siendo el valor de cada celda la intensidad del píxel. Se hace una partición de 1000 datos de test y 4000 de entrenamiento.

Los procesos de entrenamiento para encontrar las θ se harán minimizando la función de coste logístico regularizado a través del algoritmo L-BFGS.

Como métrica de evaluación de los diferentes modelos se utilizará la tasa de acierto con los datos de entrenamiento y de test. Dado que los datos se permutan aleatoriamente para cada carga de datos, es importante tener en cuenta que las métricas pueden variar ligeramente.

Modelo básico

En primer lugar, se entrena un modelo básico sin regularizar. Una vez entrenado, se puede generar una matriz de confusión para los datos de entrenamiento, que muestra la cantidad de predicciones que se ha hecho de una clase para cada clase real.



En la diagonal principal de la matriz se encuentran los datos que se han clasificado correctamente.

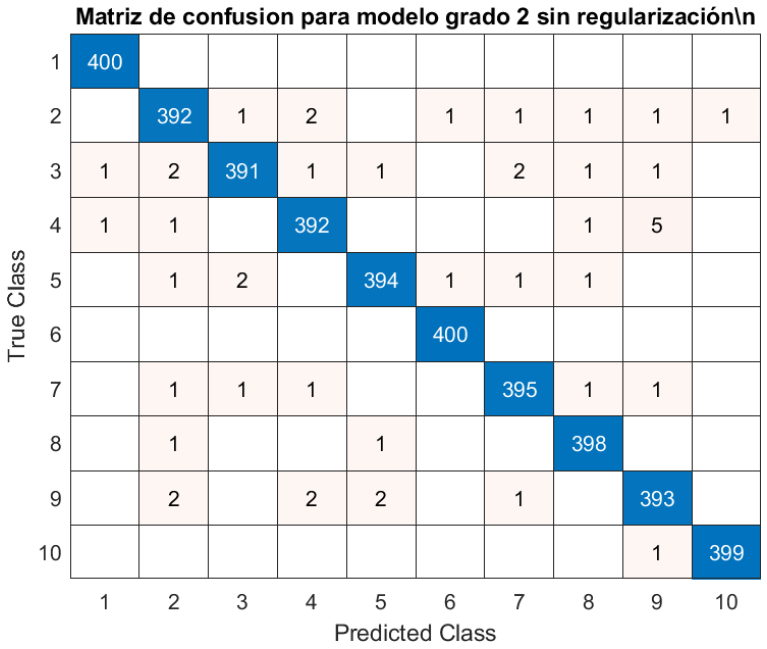
Si se hace una evaluación de la tasa de acierto se obtienen los siguientes resultados.

	Ta train	**Ta test**
Básico	0.9747	0.8680

Es interesante remarcar que la tasa de acierto para los datos de entrenamiento es notablemente mayor que para los de test, lo que puede ser un indicativo de que se puede estar produciendo sobre-ajuste.

Modelo con expansión de atributos

Para comprobar en qué medida afecta la expansión de atributos en el sobre-ajuste se ha desarrollado también un modelo que utiliza expansión polinómica de grado 10 para los datos de entrada. Los resultados de su evaluación son los siguientes.



	Ta train	**Ta test**
Log Exp grado 10	0.9892	0.8780

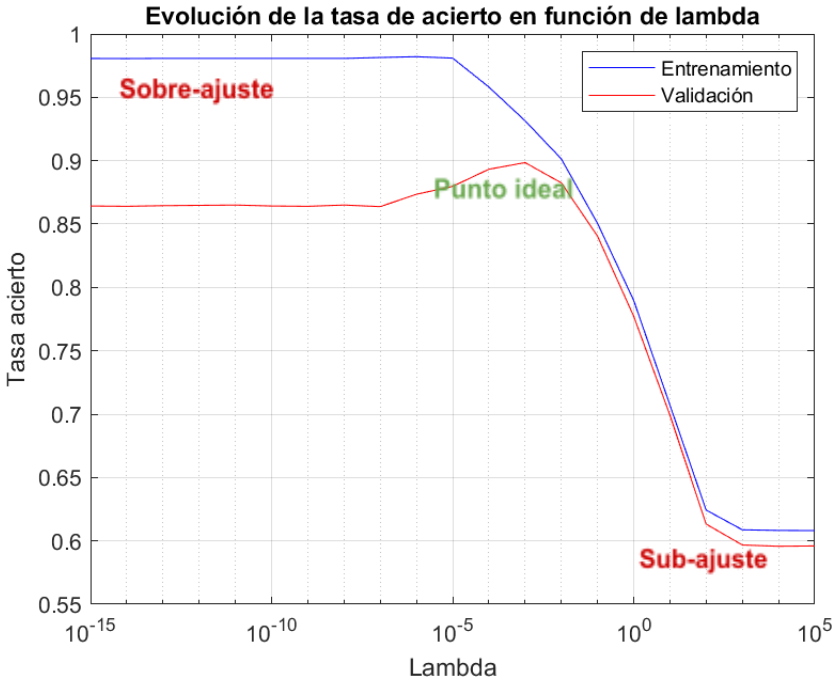
Puede ser interesante expandir porque se aumenta la tasa de acierto con los datos de test, aunque aumenta aún más el sobreajuste.

Modelo con regularización

Se va a utilizar regularización sobre los datos base (sin expandir). Para elegir el parámetro de regularización λ se va a utilizar k-fold cross-validation y se guarda la lambda que produce mayor tasa de acierto.

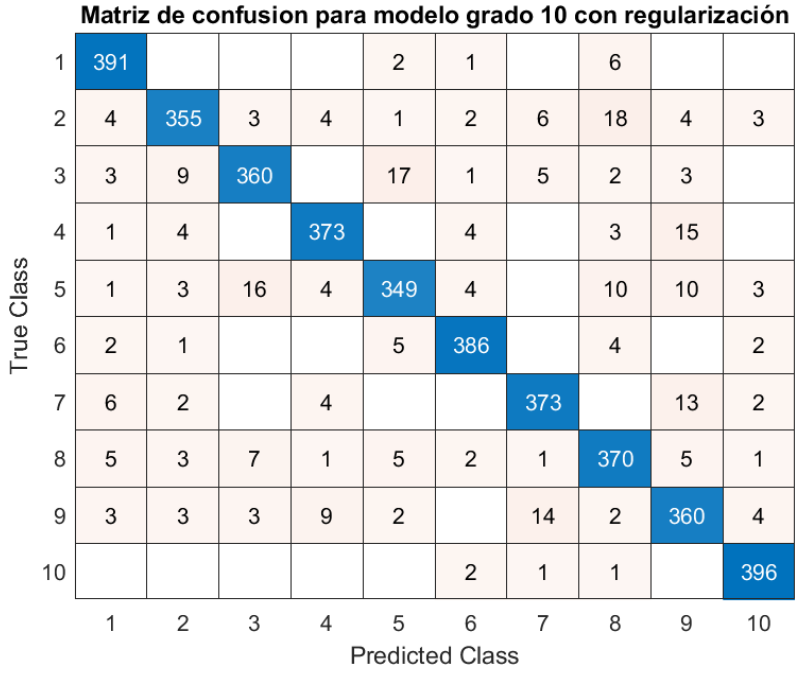
Se han probado valores de λ en progresión geométrica entre 10^{-15} y 10^{5}. Entrenando con k = 5 se ha seleccionado 10^{-3} como mejor λ, para la que se obtiene una media de Ta de 0.9315 para datos de entrenamiento y 0.8985 para los de validación.

Se puede mostrar la evolución de esta métrica en función de lambda.



Se observa que para valores altos de λ la penalización es demasiado alta y el modelo no se puede adaptar a los datos produciendo sub-ajuste. Por el contrario, para valores demasiado bajos, la regularización no es suficiente y se sigue produciendo el sobreajuste que se puede ver en el modelo básico. En el punto ideal la tasa de acierto para los datos de validación es máxima, y además muy cercana a la que se obtiene para los datos de entrenamiento.

Se entrena el modelo utilizando la mejor lambda encontrada (10^{-3}) con todos los datos de entrenamiento. Estos son los resultados.



	Ta train	**Ta test**
Regularizada	0.9282	0.8900

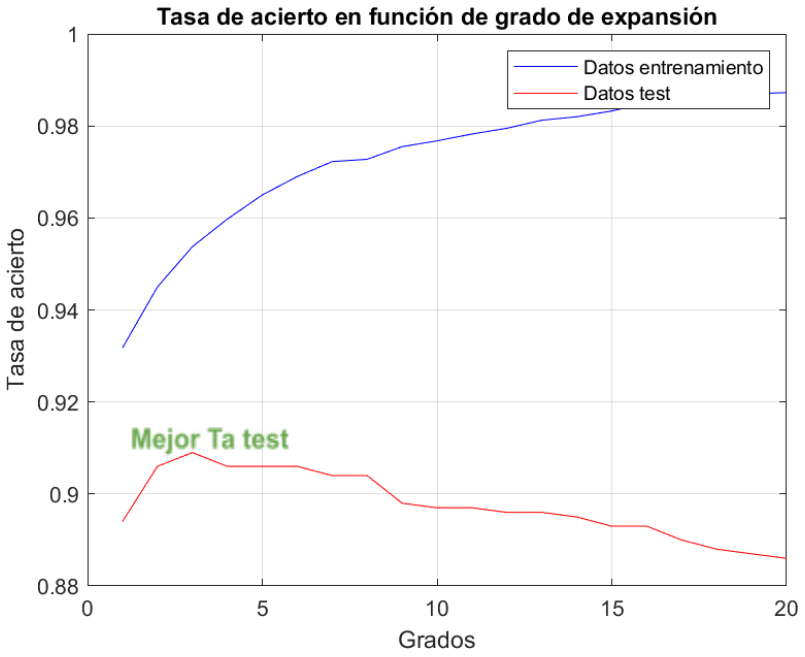
Si bien la tasa de acierto para los datos de entrenamiento ha disminuido, la de test ha aumentado, que es el indicativo real de cómo funcionará el modelo con datos nuevos. Los valores de ambas métricas son muchos más cercanos que en modelos anteriores, lo que indica que se ha eliminado el sobre-ajuste.

Se ha probado a repetir este apartado aplicando expansión de los atributos de grado 10, y el resultado ha sido que la mejor lambda es la misma (10^{-3}), aunque da lugar a peor tasa de acierto para los datos de validación durante el entrenamiento y test al entrenar con todos los datos.

Pruebas con grados

Una vez elegida una lambda, es interesante regularizar con ella probando distintos grados de expansión de polinomios y comprobar hasta qué punto se puede conseguir un modelo mejor mientras se mantiene la misma diferencia entre métricas para datos de entrenamiento y test.

Se han entrenado modelos con expansión polinómica de grados entre 1 y 20 recogiendo la tasa de acierto para los datos de entrenamiento y los de test.



En la figura se observa que utilizando $\lambda=10^{-3}$, la expansión polinómica de grado 3 es la que mayor tasa de acierto ofrece para los datos de test. Para grados más altos la tasa de acierto disminuye para los datos de test aunque aumente para los de entrenamiento. La diferencia entre tasas de acierto (*Ta_Train - Ta_Test*) para grado 1 y grado 3 es muy parecida, por lo que tendría sentido utilizar la expansión de grado 3.

Para grados más altos que 3, el sobreajuste es tan alto que la tasa de acierto de test empieza a disminuir.

Evaluación del modelo final

En este apartado se va a re-entrenar el mejor modelo con todos los datos de entrenamiento y se va a evaluar más en profundidad.

Se entrena con todos los datos de entrenamiento para obtener los siguientes resultados.

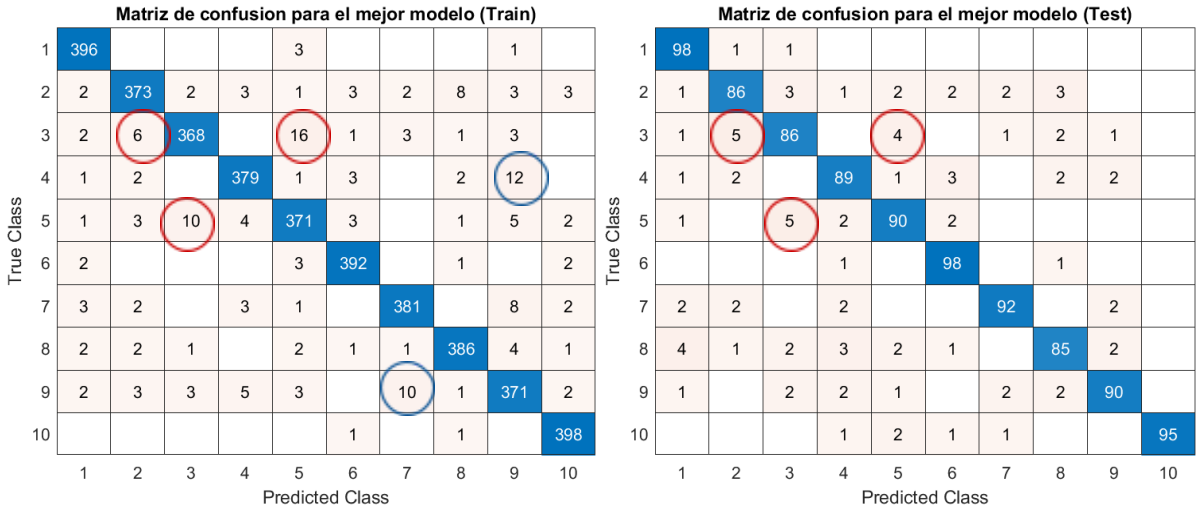
	Ta train	**Ta test**
Reg grado 3	0.9537	0.9090

Para analizar los números más problemáticos al clasificarlos, se puede calcular la precisión y el recall para cada clase. La primera métrica indicará cuántas de las muestras clasificadas como un número lo serán realmente y la segunda la proporción de muestras de una clase que efectivamente se clasifican como positivas. También se va a contar el número de muestras clasificadas como cada uno de los números para analizar si el modelo tiene tendencia hacia alguna clase en concreto. Los datos de entrada tienen una distribución perfectamente equilibrada.

	pTrain	**pTest**	**rTrain**	**rTest**	**C**
1	0.9635	0.8990	0.9900	0.9800	109
2	0.9539	0.8865	0.9325	0.8600	97
3	0.9583	0.8686	0.9200	0.8600	99
4	0.9619	0.8811	0.9475	0.8900	101
5	0.9251	0.8823	0.9275	0.9000	102
6	0.9702	0.9158	0.9800	0.9800	107
7	0.9596	0.9387	0.9525	0.9200	98
8	0.9625	0.8947	0.9650	0.8500	95
9	0.9115	0.9278	0.9275	0.9000	97
0	0.9707	1	0.9950	0.9500	95

Es interesante observar que para números como el 1 y el 6, para los cuales el modelo tiene cierta tendencia a predecir, el recall es muy alto porque casi todos los casos reales se clasifican correctamente, aunque la precisión es baja porque hay muchos falsos positivos. Para otros números como el 8 o el 0 ocurre lo contrario.

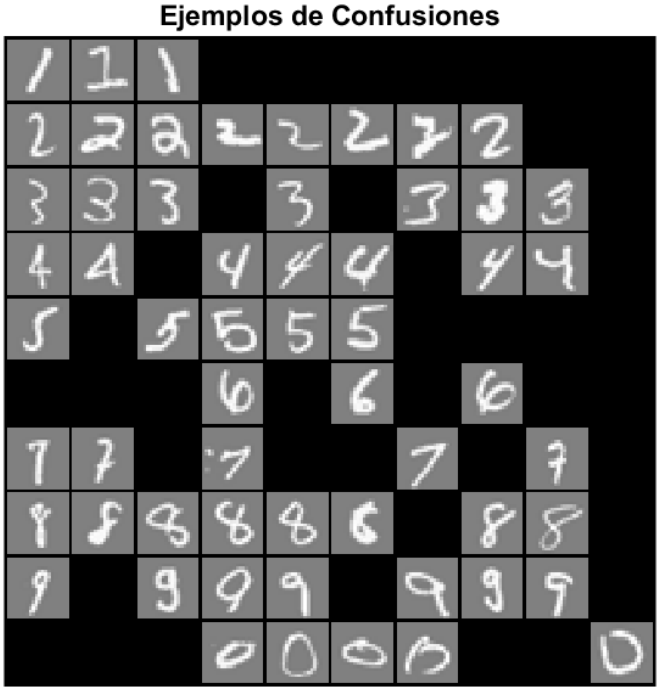
Se puede observar qué números suelen confundirse entre sí dibujando la matriz de confusión.



Se observa que tanto en los datos de entrenamiento como en los de test se tiende a confundir el número 3 con el 2 o con el 5.

Además, para los de entrenamiento hay muchos casos en los que el número 4 se confunde con el 9, y el los que el 9 se predice como 7.

Todos estos casos son números que pueden llegar a tener formas muy parecidas , por lo que tiene sentido que sean las que más fácilmente se pueden confundir. Podemos mostrar una imagen de ejemplo para cada celda de la matriz de confusión.



Las imágenes que se confunden suelen tener alguna similaridad con la clase predicha incorrectamente, por ejemplo, tiene sentido que por ejemplo los números escritos demasiado altos y delgados se puedan clasificar como unos, o que cuando son muy redondeados el modelo los pueda clasificar como cincos u ochos.

Conclusiones

A continuación se muestra un resumen de las tasas de acierto recogidas a lo largo de la práctica. Todas las métricas se han tomado con la misma permutación de los datos cargados.

	Ta train	**Ta test**
Log Básica	0.9747	0.8680
Log Exp grado 10	0.9892	0.8780
LogReg	0.9282	0.8900
LogReg Exp grado 3	0.9537	0.9090

En un problema con tantos datos de entrada se puede producir sobreajuste de base. Realizar expansión polinómica de los atributos puede llegar a mejorar los resultados a coste de aumentar aún más el sobreajuste. Los modelos regularizados ofrecen las mejores métricas.