Práctica 2

Regularización y Selección de Modelos

Javier Herrer Torres (NIP: 776609)

Aprendizaje automático
Grado en Ingeniería Informática

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

Índice

1. Objetivos

El objetivo es aplicar y comparar técnicas de regularización y validación cruzada para selección de modelos en una aplicación de regresión lineal en un caso real. Se trata encontrar el mejor modelo polinómico para predecir el precio de un coche de segunda mano en función de tres atributos: antigüedad, kilómetros y potencia.

2. Metodología

2.1. Sub-Ajuste vs. Sobre-Ajuste

Para decidir le punto ideal de complejidad de un modelo, se ha empleado la transparencia 16 del tema de Regularización.

- Sub-ajuste cuando
  - $J_{train}(\theta)$ alto, y
  - $J_{cv}(\theta) \approx J_{train}(\theta)$.
- Sobre-ajuste cuando
  - $J_{train}(\theta)$ bajo, y
  - $J_{cv}(\theta) >> J_{train}(\theta)$

2.2. Estudio previo: K-fold Cross-Validation

Para la implementación se ha seguido el algoritmo de la transparencia 13 del tema de Regularización. Además, se ha escogido como *Learner*, algoritmo de aprendizaje, la ecuación normal. El descenso de gradiente tiene demasiadas iteraciones como para obtener un resultado en un tiempo razonable.

```
function [hypothesis, best_model] = kfold_cross_validation(k, n, X, y)
best_model = 0;
best_errV = Inf;
errT_aux = 0;

% para los distintos valores de los hyper-parámetros
for model = 1:n
    err_T = 0;
    err_V = 0;
    Xexp = expandir(X, [model 1 1]);
    %separar N/k ejemplos para validación
    for fold = 1:k
        [Xcv, ycv, Xtr, ytr] = particion(fold, k, Xexp, y);
        % aprender con el resto
        h = regresion(Xtr, ytr);
        err_T = err_T + RMSE(h, Xtr, ytr);
        err_V = err_V + RMSE(h, Xcv, ycv);
    end
    % calcular el error medio de las k veces
    err_T = err_T / k;
    err_V = err_V / k;
    if  (abs(errT_aux - err_T) < UMBRAL) && (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
Xexp = expandir(X, [best_model 1 1]);
hypothesis = regresion(Xexp, y);
```

3. Resultados

3.1. Grado del polinomio para la antigüedad del coche

Se ha programado el algoritmo de k-fold cross-validation para elegir el grado del polinomio de antigüedad (entre 1 y 10), dejando fijos los kilómetros y la potencia con grado 1. En la figura 1 se encuentran dibujadas las curvas de evolución de los errores RMSE de entrenamiento y de validación.

Se puede observar que el punto ideal, en el que converge J_{train} y se obtiene además un menor valor de J_{cv} es en el **grado 5**.

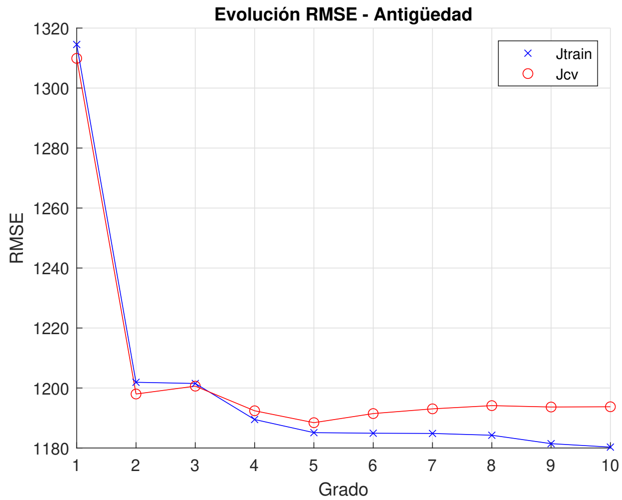

Figura 1: Evolución de los errores. Grado para la antigüedad del coche

3.2. Grado del polinomio para los kilómetros

Dejando fijo el grado 5 para la antigüedad y grado 1 para la potencia, se ha elegido mediante k-fold cross-validation el grado del polinomio de los kilómetros (entre 1 y 10). En la figura 2 se encuentran dibujadas las curvas de evolución de los errores RMSE de entrenamiento y de validación.

Se puede observar que el punto ideal, en el que converge J_{train} y se obtiene además un menor valor de J_{cv} es en el **grado 6**.

3.3. Grado del polinomio para la potencia

Dejando fijos los grados 5 para la antigüedad y 6 para los kilómetros encontrados en los apartados anteriores, se ha elegido mediante k-fold cross-validation el grado del polinomio de la potencia. En la figura 3 se encuentran dibujadas las curvas de evolución de los errores RMSE de entrenamiento y de validación.

Se puede observar que el punto ideal, en el que converge J_{train} y se obtiene además un menor valor de J_{cv} es en el **grado 6**.

Entrenamiento con el mejor modelo

Posteriormente, con el mejor modelo encontrado, se ha entrenado con todos los datos y obtenido el siguiente error:

$$RMSE_{cv} = 1009{,}32$$

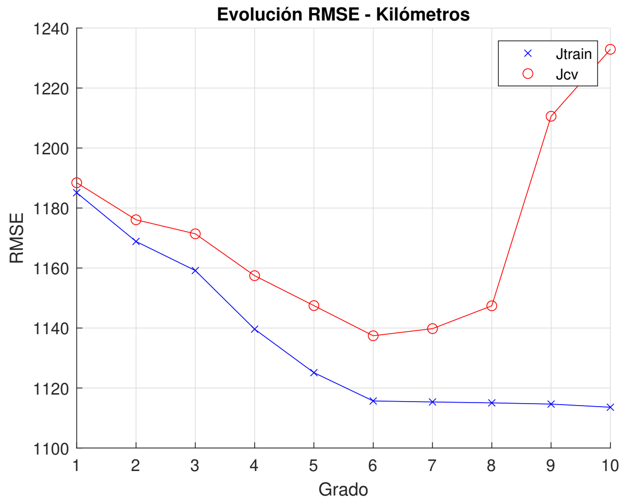

Figura 2: Evolución de los errores. Grado para los kilómetros

3.4. Regularización

Se ha realizado el ajuste de un polinomio de grado 10 para la antigüedad, los kilómetros y la potencia, utilizando regularización. Se ha elegido el parámetros lambda mediante k-fold cross-validation. En la figura 4 se encuentran dibujadas las curvas de evolución de los errores RMSE de entrenamiento y de validación

Se puede observar que el punto ideal, en el que converge J_{train} y se obtiene además un menor valor de J_{cv} es en el $\lambda = 2{,}33 \cdot 10^{-7}$.

Entrenamiento con el mejor modelo

Posteriormente, con el mejor modelo encontrado, se ha entrenado con todos los datos y obtenido el siguiente error:

$$RMSE_{cv} = 1009{,}36$$

4. Conclusiones

Los mejores grados obtenidos en los 3 primeros apartados son los siguientes:

- Antigüedad: 5
- Kilómetros: 6
- Potencia: 6

El mejor valor del parámetro de regularización λ para el último apartado es $\lambda = 2{,}33 \cdot 10^{-7}$. Además, comparando los $RMSE_{cv}$ de ambos modelos, se puede apreciar que que el modelo del apartado 3.3 es mejor.

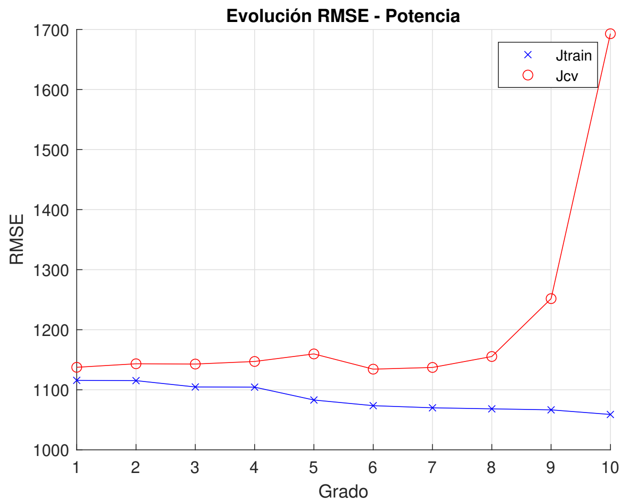

Figura 3: Evolución de los errores. Grado para la potencia

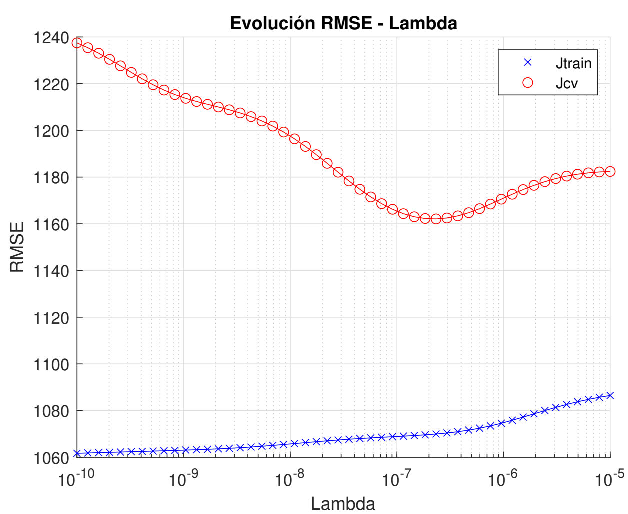

Figura 4: Evolución de los errores. Regularización