Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Métodos Numéricos

Primer cuatrimestre de 2024

Trabajo práctico 3

Cuadrados mínimos con regularización con SVD e interpolación de Legendre

Integrantes

Grupo –

Integrante	LU	Correo electrónico
Ranieri, Martina	1118/22	martubranieri@gmail.com
Rivarola, Mariana	300/19	maru.ribro@gmail.com

Resumen

Para este trabajo implementamos la función de cuadrados mínimos con y sin regularización, analizando cómo la regularización afecta los coeficientes β, que representan los parámetros que minimizan la diferencia entre los valores observados y predichos. La regularización ayuda a prevenir el sobreajuste. También exploramos la regresión polinomial usando los polinomios de Legendre y encontramos el mejor lambda y grado del polinomio al minimizar el error.

Palabras Claves

Cuadrados mínimos; Regularización; Polinomios de Legendre;

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Introducción

El objetivo de este informe es utilizar la descomposición de valores singulares[1] de una matriz para resolver el problema de cuadrados mínimos con y sin regularización, llevando a cabo su implementación para luego realizar una experimentación con estos. Además se planteó el estudio del ajuste de las curvas utilizando los polinomios de Legendre.

1.1. Cuadrados mínimos

Dado un conjunto de pares ordenados de valores (x_i, y_i) para $i = 1, \cdots, m$, buscamos una función $f(x)$ perteneciente a una familia $\mathcal{F}$ que "mejor aproxime" a los datos. El problema de cuadrados mínimos en general es:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

Si restringimos a la familia de funciones para que sea la dada por combinaciones lineales de una base con funciones linealmente independientes $\mathcal{F} = \left\{ f(x) = \sum_{j=1}^{n} \beta_j \varphi_j(x) \right\}$, podemos obtener el problema de cuadrados mínimos lineales, siendo este:

$$\min_{\beta_1, \cdots, \beta_n} \sum_{i=1}^{m} (\sum_{j=1}^{n} \beta_j \varphi_j(x_i) - y_i)^2$$

El problema se puede formular de manera matricial como:

$$\min_{\beta \in \mathbb{R}^n} \|A\beta - y\|_2^2$$

Cuando tenemos más datos que funciones (m > n) su resolución está dada a partir de las **ecuaciones normales** $A^t A\beta = A^t y$ y es:

$$\beta = ((A^t A)^{-1} A^t y) \tag{1}$$

Si conocemos la descomposición en valores singulares (SVD) de $A = USV^t$, consideramos que estamos en el caso m > n y además A tiene rango máximo (todas sus columnas son linealmente independientes), entonces se pueden reducir las matrices S y U de forma que S quede cuadrada (eliminando filas nulas) y a partir de la ecuación 1 obtenemos:

$$\beta = VS^{-1} U^t y \tag{2}$$

En la ecuación (2) no es costoso invertir la matriz ya que S es una matriz diagonal. Una vez que tenemos el valor de β que minimiza, el error total del cálculo está dado por:

$$ECM = \|A\beta - y\|_2^2 \tag{3}$$

1.2. Cuadrados mínimos con regularización

Un problema que puede surgir de utilizar el método de la sección anterior es que los parámetros de β pueden tomar un valor arbitrariamente grande. Para evitar esto, se puede usar regularización L2 (cuadrados), también conocida como regresión Ridge. La idea consta en incluir un nuevo término que penalice coeficientes muy grandes en lo que queramos minimizar. Logramos esto con un nuevo parámetro λ, el cual nos indica cuán fuerte es esta penalización. La importancia de regularizar se verá a mayor detalle en **resultados**.

$$\min_{\beta_1, \ldots, \beta_n} \sum_{i=1}^{m} (\sum_{j=1}^{n} \beta_j \varphi_j(x_i) - y_i)^2 + \lambda \sum_{j=1}^{n} \beta_j^2 \tag{4}$$

Si derivamos con respecto a β y se iguala la ecuación (4) a cero, quedando $(A^t A + \lambda I)\beta = A^t y$.
Haciendo el procedimiento análogo a la sección anterior llegamos a:

$$\beta = (A^t A + \lambda I)^{-1} A^t y \tag{5}$$

Esta expresión también puede ser planteada en términos de SVD, obteniendo:

$$\beta = VS(S^2 + \lambda I)^{-1} U^t y \tag{6}$$

[1] SVD

1.3. Regresión polinomial con polinomios de Legendre

Cuando utilizamos una base de funciones linealmente independientes basada en polinomios para resolver este problema, decimos que se hace regresión polinomial. En este trabajo se utilizarán los polinomios de Legendre, que tienen un número de condición bajo y forman además una base ortonormal.

Los polinomios de Legendre $P_n(x) = 1$ son una secuencia de polinomios ortogonales en el intervalo [-1, 1] que se definen mediante la relación de recurrencia (es decir, dado el polinomio de Legendre de grado n y n - 1, podemos calcular el de grado n + 1):

- Los polinomios base: $P_0(x) = 1, P_1(x) = x$
- Para grados superiores: $(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$

1.4. Sobreajuste

Una vez encontrada la solución al problema de cuadrados mínimos con una muestra A, queremos ver qué tan bien aproximan las soluciones de un nuevo conjunto de datos de la misma distribución. Si el grado de polinomio es muy alto o no se usa regularización, es probable que el error para la nueva muestra sea elevado. Esto se conoce como **sobreajuste**, una solución posible para encontrar los valores óptimos de grado, y λ es la optimización de hiperparámetros con validación cruzada: Se divide la muestra original en datos de ajuste y datos de validación. Se consigue β a partir de los primeros y se busca minimizar el error cuadrático de los segundos de la siguiente forma:

$$\beta = (A_{ajuste}^t A_{ajuste} + \lambda I)^{-1} A_{ajuste}^t y_{ajuste}$$
$$ECM_{ajuste} = \|A_{ajuste}\beta - y_{ajuste}\|_2^2$$
$$ECM_{val} = \|A_{val}\beta - y_{val}\|_2^2$$

2. Desarrollo

2.1. Derivación de cuadrados mínimos

De la ecuación (5) podemos utilizar la descomposición SVD de A para obtener una ecuación más conveniente, más que nada a la hora de calcular la inversa. Para esto, primero reemplazamos a A por su descomposición SVD.

$$\beta = ((USV^t)^t USV + \lambda I)^{-1}(USV^t)^t y$$
$$\beta = (VSU^t USV^t + \lambda I)^{-1} VSU^t y$$

Como U es una matriz ortogonal, $U^t = U^{-1}$ entonces $UU^t = I$. Análogamente a la ecuación (5), se reducen U y S para que S quede cuadrada, eliminando las filas nulas. De este modo tenemos que S es diagonal y cuadrada, entonces $S = S^t \longrightarrow SS = S^2$

$$\beta = (VSISV^t + \lambda I)^{-1} VSU^t y$$
$$\beta = (VSSV^t + \lambda I)^{-1} VSU^t y$$
$$\beta = (VS^2V^t + \lambda I)^{-1} VSU^t y$$

Como V es una matriz ortogonal, $V^t = V^{-1}$ entonces $VV^t = I$. Podemos sacar factor común V y V^t.

$$\beta = (V(S^2 + \lambda I)V^t)^{-1} VSU^t y$$
$$\beta = ((V^t)^{-1}(S^2 + \lambda I)^{-1}V^{-1}) VSU^t y$$
$$\beta = V(S^2 + \lambda I)^{-1} V^t VSU^t y$$
$$\beta = V(S^2 + \lambda I)^{-1} ISU^t y$$

Como $S^2 + \lambda I$ es suma de matrices diagonales cuadradas, el resultado es una matriz diagonal cuadrada. Como $(S^2 + \lambda I)^{-1}$ y S son ambas diagonales, causando que conmuten. Finalmente tenemos lo mismo que la ecuación (6):

$$\beta = VS(S^2 + \lambda I)^{-1} U^t y$$

2.2. Funciones auxiliares generales para los algoritmos

- $svd(A)$ Calcula la descomposición SVD de A. En Python es np.linalg.inv [2]
- $diagonal(A)$ Devuelve la Diagonal de A. En Python esnp.diag [3]

[2] numpy-linalg-svd
[3] esnp.diag

- *traspuesta*(A) Calcula la transpuesta de A. En Python es numpy.ndarray.T [4]
- *legendre*$(A, grado)$ Calcula el polinomio de Legendre de grado indicado. [5]

En nuestro contexto tenemos que invertir únicamente matrices diagonales, por eso utilizamos la descomposición SVD, así a la hora de calcular la inversa de la matriz diagonal es $1/s_i$, s_i elemento de la diagonal que queremos invertir.

2.3. Cuadrados mínimos lineales

Para realizar el algoritmo utilizamos la descomposición SVD de la matriz y utilizamos la Ecuación 2.

Algorithm 1 CML

```
Parámetros: A, y
U, S, Vt = svd(A)                          ▷ Calculamos la descomposición SVD
S_inversa = diagonal(1 / S)
beta = traspuesta(Vt) * S_inversa * traspuesta(U) * y
return beta
```

2.4. Cuadrados mínimos lineales con Regulación

Para realizar el algoritmo utilizamos la descomposición SVD de la matriz y utilizamos la Ecuación 2.

Algorithm 2 CML Regulación

```
Parámetros: A, y, lambda
U, S, Vt = svd(A)                          ▷ Calculamos la descomposición SVD
S_diagonal = diagonal(S)
S_diag = diagonal(1 / (S_reducida**2 + lambda))
beta = traspuesta(Vt) * S_diagonal * inv_diag * traspuesta(U) * y
return beta
```

3. Resultados

3.1. Cuadrados mínimos lineales sin y con regularización

En este experimento, generamos una matriz aleatoria de tamaño $100x10$ utilizando una función de generación de números aleatorios 'rand'[6], la cual produce números aleatorios siguiendo una distribución uniforme entre 0 y 1. Esta matriz se utiliza para calcular 10 coeficientes β y observar los efectos de la regularización en el método de mínimos cuadrados. La regularización tiende a penalizar coeficientes más grandes mediante la incorporación de un término adicional en la función de cuadrados mínimos, lo que hace que los coeficientes β sean más estables y menos propensos a fluctuaciones extremas.

Cuando usamos cuadrados mínimos lineales sin regularización, los coeficientes β varían en magnitud, mostrando valores más dispersos. Esto se observa con mayor claridad cuando la matriz tiene menos muestras en comparación con el número de coeficientes β, como se muestra en las lineas azules de la Figura 1. Por otro lado, al aplicar regularización (regresión Ridge), los coeficientes β tienden a ser más suaves y menos sensibles a cambios. A medida que aumentamos el valor de λ (el parámetro de regularización) la penalización de coeficientes grandes se intensifica, lo que resulta en coeficientes más cercanos a cero y una distribución más uniforme.

Además de estabilizar los coeficientes β, la regularización tiene beneficios en la capacidad predictiva de un modelo. Al limitar la magnitud de los coeficientes β, la regresión ridge ayuda a prevenir el sobreajuste, donde el modelo se adapta demasiado a los datos de entrenamiento y pierde capacidad de generalización a nuevos datos. Esto es útil cuando el número de muestras es pequeño en comparación con el número de características, ya que la regularización permite obtener estimaciones más fiables y menos sensibles a variaciones aleatorias en los datos. [1]

[4] numpy.ndarray.T
[5] numpy.polynomial.legendre.Legendre
[6] numpy.random.rand

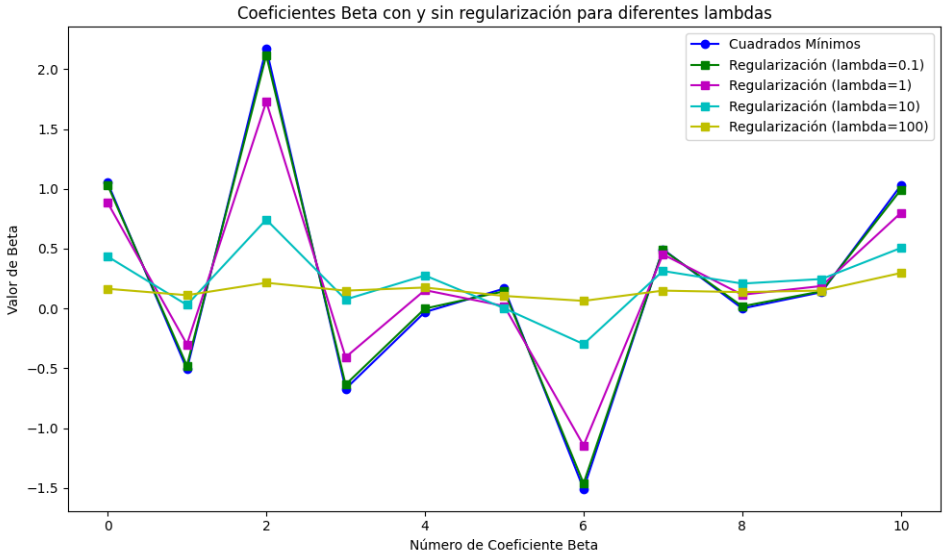

Figura 1: Gráfico de los valores de beta en función de distintos valores de lambda

3.2. Errores de ajuste y de validación

Para este experimento se nos pidió ajustar una combinación lineal de polinomios de Legendre (sin regularización) en los datos de ajuste y predecir los valores de validación. Contamos para ello con dos conjuntos de datos, uno de ajuste y otro de validación.

Primero tuvimos que calcular los coeficientes β para los polinomios de Legendre de diferentes grados utilizando cuadrados mínimos. Para esto utilizamos el algoritmo 1. Para cada grado calculamos el error de ajuste y validación.

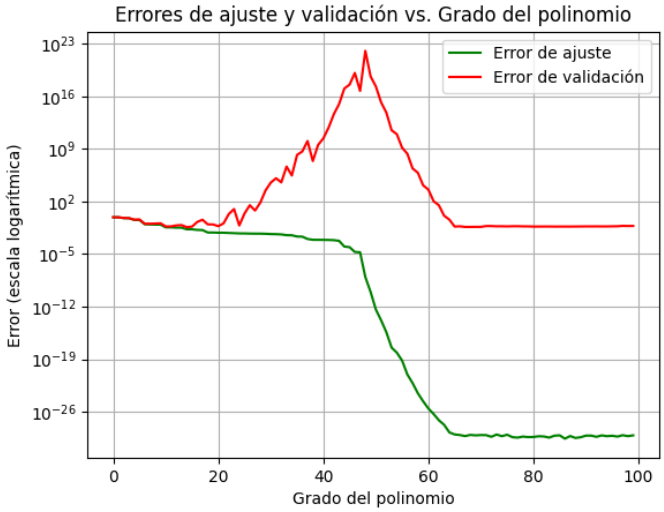

Figura 2: Gráfico con los errores de ajuste y validación.

El gráfico que muestra el error de validación inicialmente presenta un descenso del error, indicando que el polinomio está capturando mejor los datos (grados 1-15). Luego empieza a aumentar el error numérico, y mirando la escala del gráfico podemos notar un significativo aumento ya que pasamos de estar de un error entre $10^{-5} - 10^{2}$ hasta un pico de 10^{23} en el grado 50. Este aumento nos indica que el polinomio está sobreajustando los datos de entrenamiento, capturando las oscilaciones de los datos.

Vemos un segundo descenso en el error de validación en los grados 50-65 para luego mantenerse con un error estable en los grados 65-100, lo que nos indica que el modelo más complejo generaliza mejor los datos. Igualmente, podemos observar que comparado a los grados 1-15 no tiene un error menor, si no que son aproximadamente similares. El mínimo error obtenido en validación fue ≈ 0.037 en el grado 14. En el error de validación se puede ver el fenómeno conocido como **doble descenso**. Este fenómeno se ve en estadística y machine learning y ocurre cuando un modelo estadístico cuenta con escasa o excesa cantidad de parámetros, y tienen la particularidad de que ambos tienen un error muy pequeño (en nuestro caso los parámetros son los grados del polinomio). Pero un modelo con una cantidad de parámetros aproximadamente igual que la cantidad de datos utilizados para entrenar un modelo tienen un error mucho mayor (en nuestro caso los datos son los 50 puntos).

Ahora, si observamos el error de ajuste, podemos ver que cuando el grado del polinomio es bajo, el polinomio no puede capturar muy bien los datos, lo que equivale a un error de ajuste relativamente alto. A medida que aumenta el grado, el modelo tiene más flexibilidad para ajustarse mejor a los puntos de datos, lo que resulta en una reducción gradual del error. Después del grado 50, podemos observar una disminución aún mayor en el error de ajuste, casi llegando al cero. Esto ocurre ya que el grado del polinomio es mayor a la cantidad de datos, entonces captura las fluctuaciones más sutiles de nuestro conjunto.

Con todo esto, podemos decir que es importante tener en cuenta que mientras el error de ajuste puede seguir disminuyendo, el error de validación puede comenzar a aumentar drásticamente después de cierto grado. Esto indica un posible sobreajuste, pues el modelo se ajusta demasiado bien a los datos de entrenamiento, pero pierde en capacidad para generalizar nuevos datos. Por esto es importante considerar tanto el error de ajuste como el error de validación para evaluar la capacidad predictiva del modelo en datos no vistos. Un equilibrio entre la complejidad del modelo (grado) y su capacidad para generalizar (representado por el error de validación) es importante para obtener buenos resultados en la práctica.

3.3. Búsqueda de mejor lambda y grado de polinomio de Legendre

Para poder explorar los hiperparámetros del grado y lambda, elegimos el rango inicial de grados del 1 al 80 ya que, por lo visto en la anterior sección, aproximadamente en 70 el error se estabiliza y no tiene mucho sentido explorar en polinomios de grados mayores. Para los λ exploramos en el rango 0 a 1 ya que con este rango podemos ver un ajuste gradual en el ECM, ayudándonos a entender mejor el impacto de la regularización.
Probamos 100 valores de lambda espaciados uniformemente.

Luego de explorar en esos rangos, obtuvimos como resultados:

- Lambda: 0.02020
- Grado: 32
- Mínimo ECM de ajuste: 0.01305

Como podemos ver en el gráfico, con grado $= 32$ y $\lambda \approx 0.2$, cometemos un error bastante bajo. Se aprecia que nuestra predicción se asemeja bastante a los datos. El grado del polinomio nos indicaba qué tan compleja es la función para poder aproximar los datos, y esta no resultó ser tan elevada. Luego, el lambda medía cuánta regularización necesitábamos para prevenir el sobreajuste, entonces al obtener este resultado, podemos decir que los datos no tienen mucha dispersión.

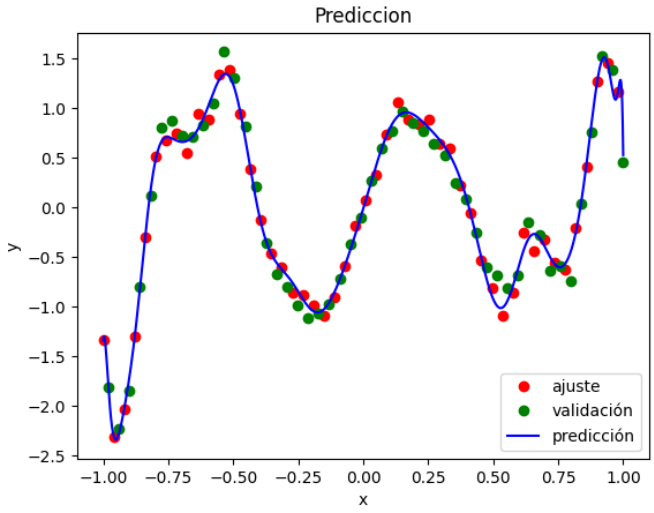

Figura 3: Imagen de la predicción a la que llegamos con los mejores λ y grado encontrados

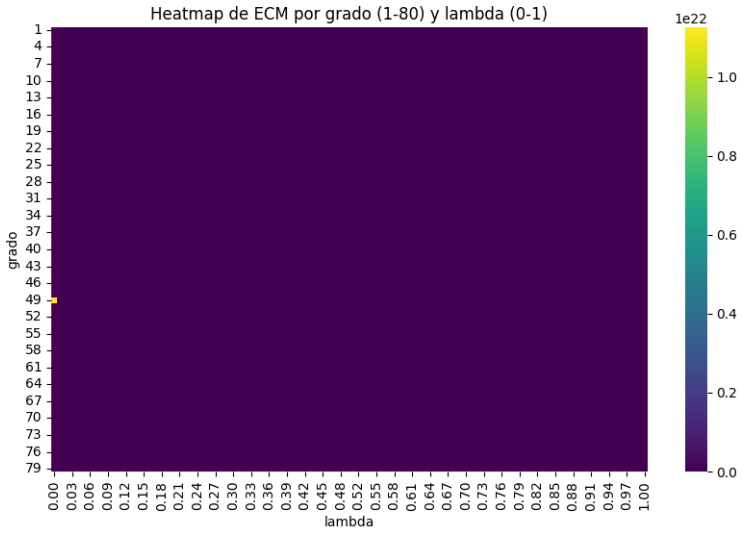

Figura 4: Heatmap con grado entre 1-80 y lambda entre 0-1

En la imagen del Heatmap inicial (Figura 4) con todos los rangos del experimento, no podemos apreciar el resultado que obtuvimos. Por eso acotamos los rangos del gráfico para poder ver mejor representados los valores obtenidos. El primer

recorte fue entre los grados 10-50 dado que tenemos 50 datos y sabemos que con los polinomios de grados mayores a nuestra cantidad de datos, nos lleva a un sobreajuste. Esto es porque el polinomio actual se ajusta perfectamente a los puntos conocidos, pero a la hora de predecir no es muy eficiente. Por otro lado, el lambda lo acotamos entre 0.01-0.5 ya que de 0.5-1 son valores muy lejanos al resultado que obtuvimos.

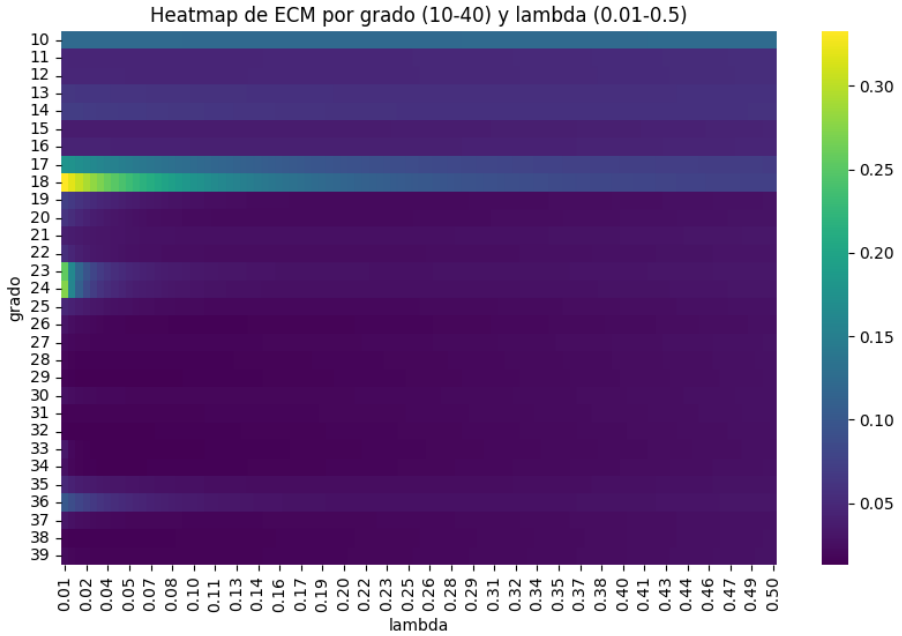

Figura 5: Heatmap con grado entre 10-50 y lambda entre 0.01-0.5

Con este Heatmap (Figura 5), si miramos la escala podemos apreciar cómo en este rango tenemos un error mucho menor que en el anterior, pero sería ideal poder llegar a una escala más chica dado que el error mínimo del experimento nos dio ≈ 0.013. Mirando el gráfico vemos que en el rango aproximado 27-34 tenemos menor error, entonces acotamos otra vez los grados pero con esos valores. El lambda también lo podemos acotar un poco más considerando que nos dio ≈ 0.02, ahora lo acotamos entre 0.01-0.2.

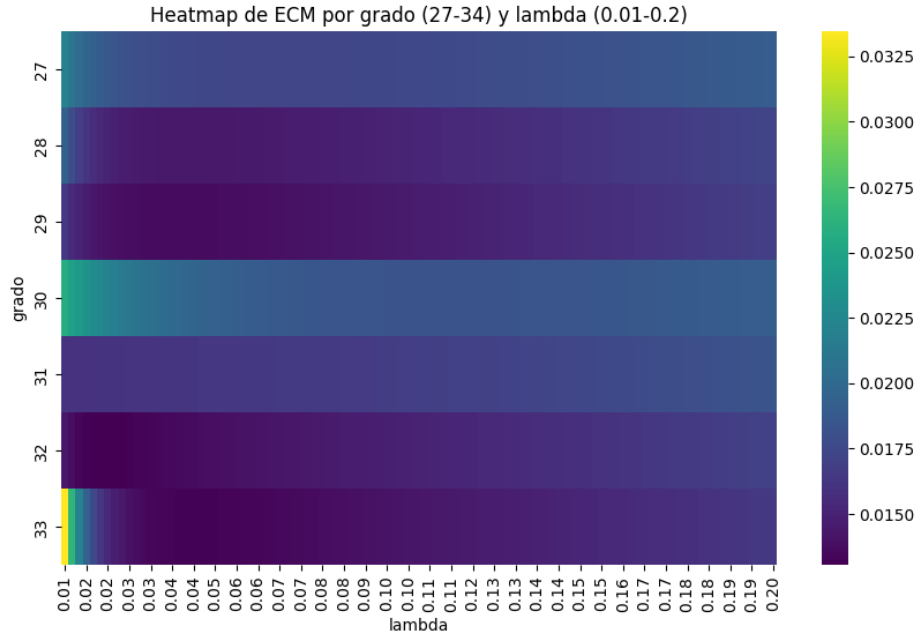

Figura 6: Heatmap con grado entre 27-34 y lambda entre 0.01-0.2

Con este recorte en los rangos, mirando la escala apreciamos que el error es aún menor y se observa que el grado 32 es levemente más oscuro que los otros, pero para visualizarlo mejor podemos hacer el mismo Heatmap graficando hasta la media del error en este rango.

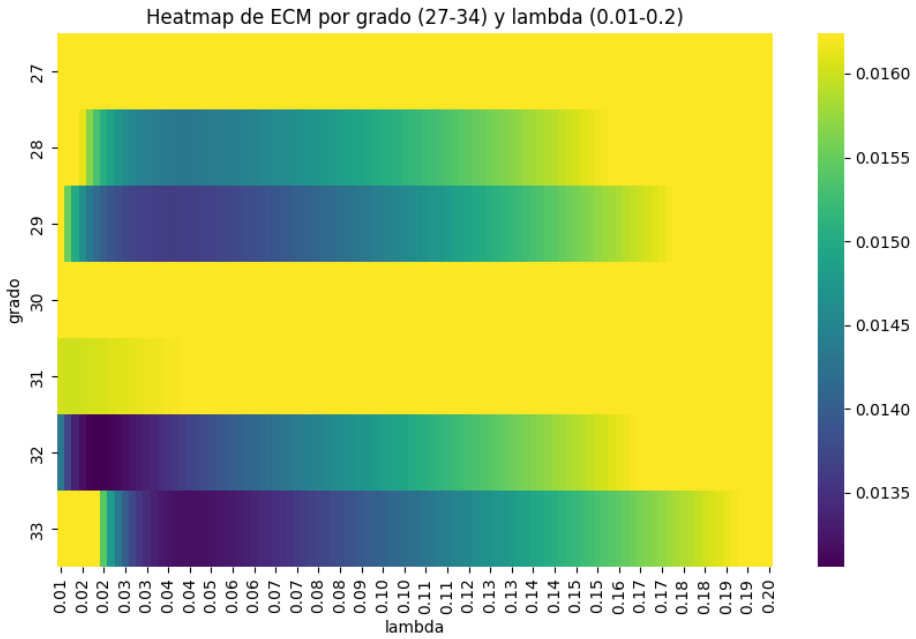

Figura 7: Heatmap con grado entre 27-34 y lambda entre 0.01-0.2 y valor máximo la media

Con este último Heatmap podemos ver mejor representado el mínimo error que obtuvimos. Podemos apreciar que el grado 32 y 33 parecen tener un mínimo error muy similar, y resulta que ser que el grado 33 con lambda ≈ 0.04 tiene un error de 0.013209, asemejándose al obtenido con el experimento pero dado que el error que tenemos con grado 32 y lambda ≈ 0.02 es de 0.01305, este último tiene un error menor.

4. Conclusiones

Después de realizar los diferentes experimentos, llegamos a varias conclusiones importantes. Primero descubrimos que los cuadrados mínimos sin regularización pueden ser ineficaces, ya que no controlan la magnitud de los coeficientes β, lo que puede llevar a un modelo que se ajuste demasiado a los datos de entrenamiento y no generalice bien a nuevos datos. Al introducir un parámetro lambda adecuado para penalizar los coeficientes beta, logramos controlar mejor este problema, pues encontramos los hiperparámetros óptimos, tanto el grado del polinomio de Legendre como el valor de lambda y mejoramos la generalización a nuevos datos, optimizando así la precisión y capacidad predictiva de nuestros modelos.

Además, observamos un fenómeno interesante en el error de validación en función del grado del polinomio de Legendre, el doble descenso. Inicialmente, al aumentar el grado, el error de validación disminuye capturando mejor la información de los datos, luego aumenta debido al sobreajuste, y eventualmente disminuye de nuevo a medida que el modelo tiene suficiente capacidad para capturar las complejidades de los datos. En ese experimento también habíamos obtenido que el mínimo error que podíamos cometer era ≈ 0.037 en el grado 14 mientras que en el experimento para encontrar el mejor grado y λ llegamos a un error ≈ 0.01305 con $\lambda \approx 0.02$ y grado 32. Esto demuestra que al utilizar regularización se obtiene un error menor ya que tiene una capacidad de generalización por lo tanto se puede adaptar mejor a datos no vistos y/o nuevos.

Resumiendo; el uso de la regularización mediante SVD y la correcta selección de hiperparámetros son cruciales para mejorar la precisión y la capacidad de generalización de los modelos de regresión basados en polinomios de Legendre, evitando el sobreajuste y capturando de manera efectiva las complejidades de los datos.

Referencias

[1] URL: https://www.linkedin.com/pulse/parte-3-regrepedia-tipos-de-regresiones-desde-las-y-mora-caballero/ (visitado 02-07-2024).