

Machine Learning Automatizado con Aplicaciones a Modelos Predictivos Reducidos

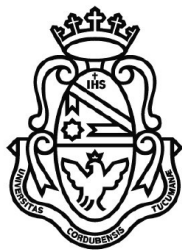
por
Franco Cerino

Trabajo presentado como parte de los requerimientos para
obtener el título de grado de

Licenciado en Física

de la

UNIVERSIDAD NACIONAL DE CÓRDOBA



© FAMAF - 2020

Director: Dr. Manuel Tiglio

Esta obra está bajo una licencia Creative Commons
"Reconocimiento-NoCommercial 4.0 Internacional".



Agradecimientos

A mi director Manuel Tiglio,
mi familia y mis amigos,
que han sido de gran ayuda a lo largo de la carrera.

Resumen

La detección directa de ondas gravitacionales a través de interferómetros, generadas por la coalescencia de un sistema binario, es uno de los descubrimientos más relevantes de la ciencia en los últimos años. Esto genera la necesidad de obtener expresiones de ondas gravitacionales de forma rápida y precisa para diferentes aplicaciones. Se pueden obtener simulaciones a través de relatividad numérica, pero la obtención de una simulación implica la utilización de supercomputadoras, demandando desde semanas hasta meses de tiempo de cómputo por simulación, lo cual implica que generar un conjunto amplio de soluciones en tiempo real es inviable a través de un enfoque directo. Para reducir este costo computacional se ha desarrollado un método predictivo sustituto para computar expresiones de ondas gravitacionales de forma precisa y rápida, aplicando teoría de aproximaciones cuasi-óptima para construir modelos reducidos para sistemas parametrizados, usando sólo un número relativamente chico de simulaciones con supercomputadoras como espacio de entrenamiento. A través de él se pueden construir soluciones correspondientes a un espacio de parámetros definido, dado por las masas y spins de los agujeros negros, en el orden de un milisegundo en una computadora personal, resultando en un factor de aceleración de al menos 11 órdenes de magnitud, sin pérdida de precisión, respecto de simulaciones por supercomputadoras. Para obtener el modelo predictivo se deben realizar regresiones sobre el espacio de parámetros en cuestión, donde la incerteza de éstas es el factor que más contribuye al error de las predicciones. En este trabajo se construyó una familia de modelos sustitutos para la colisión de dos agujeros negros y se realizó un estudio comparativo del poder predictivo conseguido a través de una gran (alrededor de 100) variedad de modelos provenientes del ámbito de Machine Learning y de métodos clásicos de aproximación, realizando regresiones globales y locales. Para posibilitar la aplicación de un amplio espectro de modelos de Machine Learning de diferente naturaleza, se utilizó la plataforma de Machine Learning Automatizado DataRobot.

Abstract

The direct detection of gravitational waves through interferometers, generated by the coalescence of a binary system, is one of the most relevant discoveries in science in recent years. This generates the need to obtain gravitational wave expressions quickly and accurately for different applications. Simulations can be obtained through numerical relativity, but obtaining a simulation implies the use of supercomputers, demanding from weeks to months of computing time per simulation, which implies that generating a wide set of solutions in real time is not feasible through a direct approach. To reduce this computational cost, a surrogate predictive method has been developed to compute gravitational wave expressions accurately and quickly, applying quasi-optimal approximation theory to build reduced models for parameterized systems, using only a relatively small number of simulations with supercomputers as training space. Through it, solutions can be constructed corresponding to a defined parameter space, given by the masses and spins of the black holes, in the order of one millisecond in a personal computer, resulting in an acceleration factor of at least 11 orders of magnitude, without loss of precision, with respect to simulations by supercomputers. To obtain the predictive model, regressions must be performed on the parameter space used, where the uncertainty of these is the factor that most contributes to the error of the predictions. In this work, a family of substitute models for the collision of two black holes was built and a comparative study of the predictive power achieved through a great (around 100) variety of models from the field of Machine Learning and classical methods was carried out, performing global and local regressions. To allow the application of a wide spectrum of Machine Learning models of different nature, the Automated Machine Learning platform DataRobot was used.

Índice general

| | |
|--|-----------|
| 1. Introducción y Motivaciones | 2 |
| 1.1. Representación de ondas gravitacionales | 3 |
| 1.2. Modelo Predictivo | 4 |
| 1.3. Enfoque del trabajo | 4 |
| 2. Marco Teórico | 6 |
| 2.1. Problema de cuadrados mínimos (Least Squares) | 7 |
| 2.2. Representación de un conjunto de elementos: error colectivo | 8 |
| 2.3. Métodos espectrales | 8 |
| 2.4. Distancia de Kolmogorov | 10 |
| 2.5. Bases Reducidas | 11 |
| 2.6. Algoritmos Greedy o Voraces | 12 |
| 2.7. Algoritmo Greedy para obtener una Base Reducida | 13 |
| 2.7.1. Convergencia del Algoritmo Greedy | 15 |
| 2.8. Interpolación polinomial | 16 |
| 2.8.1. Splines | 17 |
| 2.8.2. Convergencia: Constante de Lebesgue | 18 |
| 2.8.3. Representación vs. predicción | 18 |
| 2.8.4. Paso de Proyección a interpolación | 19 |
| 2.9. Método de Interpolación Empírica (EIM) | 19 |
| 2.9.1. Poder predictivo del algoritmo EIM | 20 |
| 2.10. Modelo predictivo a partir del interpolante empírico | 21 |
| 3. Metodología y Resultados | 23 |
| 3.1. Datos utilizados | 23 |
| 3.2. Algoritmo Greedy | 25 |
| 3.3. Algoritmo EIM | 27 |
| 3.4. Fase de regresión | 28 |
| 3.4.1. Regresiones con Machine Learning Automatizado | 30 |
| 3.4.2. Regresiones con modelos clásicos | 35 |
| 4. Conclusiones | 52 |
| 5. Apéndice | 58 |
| 5.1. Modelos utilizados por DataRobot | 58 |

Capítulo 1

Introducción y Motivaciones

La detección directa de ondas gravitacionales a través de interferómetros, generadas por la coalescencia de un sistema binario es uno de los descubrimientos más relevantes de la ciencia en los últimos años. Esto genera la necesidad de obtener expresiones de ondas gravitacionales de forma rápida y precisa para diferentes aplicaciones, como puede ser la inferencia de parámetros de una onda gravitacional medida [5] [11] [12] o la construcción de catálogos [18].

En el caso de un sistema binario de agujeros negros, para descubrir la evolución en el tiempo del sistema es necesario resolver numéricamente las ecuaciones de Einstein. El espacio de parámetros intrínseco al sistema físico es de 7 dimensiones, correspondientes al cociente de las masas de los agujeros negros y a dos vectores de spin, asociados al momento angular de cada uno.

Para realizar inferencia de parámetros de la fuente de ondas gravitacionales se necesita obtener soluciones en tiempo real, en demanda, y con métodos Bayesianos se necesitan al menos millones de ellas. Se pueden obtener simulaciones a través de relatividad numérica (NR), pero la obtención de una simulación implica la utilización de supercomputadoras [6] [10] [24], demandando desde semanas hasta meses de tiempo de cómputo *por simulación*, lo cual implica que generar un conjunto amplio de soluciones en tiempo real es inviable. Por ejemplo, si se quiere obtener un conjunto de soluciones con 100 parámetros definidos por dimensión, implica computar 10^9 soluciones de NR.

Para reducir el costo computacional de obtener soluciones se ha desarrollado un método en particular para computarlas de forma precisa y rápida. Dicho método se basa en una descomposición online-offline del problema, donde en la parte online se prepara el modelo para que pueda otorgar soluciones en el espacio de parámetros de interés y en la parte offline se realizan evaluaciones del modelo para obtener las expresiones de las ondas gravitacionales. Para realizar esto se utiliza el potencial del enfoque de bases reducidas [8]. A partir de la redundancia existente en el espacio de parámetros, se sugiere que la cantidad de información necesaria para realizar una descripción de las soluciones puede ser reducida considerablemente. Esta reducción puede ser realizada identificando un conjunto representativo de n soluciones y construyendo una base a partir de este, que permita representar a todas las soluciones a través proyecciones sobre el espacio generado por las base. Únicamente a partir de esta base, con cada elemento elegido de manera cuasi-óptima con respecto a la medida de Kolmogorov, se construye un modelo predictivo de todas las soluciones en el espacio de parámetros en cuestión. Para realizar esto se obtiene un interpolante empírico [4] [15] a partir de la base, que será el que otorgará las predicciones, en nodos elegidos también de forma cuasi-óptima, aminorando el crecimiento de la constante de Lebesgue con la cantidad de elementos de la base (que por construcción es el mismo que la cantidad de nodos temporales en las cuales se hacen un subsampleo). Para poder obtener expresiones

de los interpolantes en todos el espacio de parámetros, se realizan regresiones en dicho espacio para cada nodo interpolante. A partir de estos procedimientos se obtiene un modelo sustituto para la familia de soluciones en cuestión [19] [9], con un bajo costo de evaluación $\mathcal{O}(nL + nc_{fit})$, con L el número de muestras en el tiempo de cada serie temporal y c_{fit} el costo de evaluación de las regresiones realizadas en el espacio de parámetros.

1.1. Representación de ondas gravitacionales

Un sistema binario de agujeros negros [22] [14] se caracteriza con 7 parámetros intrínsecos: $q = m_1/m_2$, donde m_1 y m_2 son las masas correspondientes de los agujeros negros y 2 vectores de spin \vec{S}_1 y \vec{S}_2 , que describen la rotación sobre sus propios ejes. La masa total del sistema se define como $M = m_1 + m_2$. Un agujero negro se caracteriza totalmente a través de su masa m , su spin \vec{S} y su carga eléctrica Q , que no es relevante desde un punto de vista astrofísico. El producto cartesiano de todas estas variables da lugar al espacio de parámetros λ correspondiente a un sistema binario de agujeros negros.

Este trabajo se enfoca en sistemas binarios de agujeros negros inicialmente sin spin (y sin carga eléctrica), inicialmente en órbita cuasi-circular. De esta forma, el sistema se puede definir a través de un parámetro unidimensional, que es el cociente de las masas q .

En un sistema binario de agujeros negros de este tipo, éstos se atraen gravitacionalmente hasta ocurrir una colisión, de la cual se obtiene un agujero negro remanente, que se caracteriza por su masa m_f , spin \vec{S}_f y velocidad de retroceso \vec{v}_f . En el sistema se pueden identificar tres partes diferentes: *in-spiral*, *merger* y *ringdown*. El término *merger* deviene del intervalo temporal donde se produce la colisión en el sistema, luego *in-spiral* y *ringdown* identifican a la parte anterior y posterior de dicho evento, respectivamente.

Para un sistema binario de agujeros negros, las ondas gravitacionales pueden ser expresadas como

$$h(\lambda; t, r, \theta, \phi) = \sum_{l=2}^{\infty} \sum_{m=-l}^l {}_{-2}Y^{lm}(\theta, \phi) h^{lm}(\lambda; t, r),$$

donde ${}_{-2}Y^{lm}$ son los armónicos esféricos con peso de spin $s = -2$. Para distancias lejanas de la fuente se puede expresar como

$$h(\lambda; t, r, \theta, \phi) \approx \frac{1}{r} \sum_{l=2}^{\infty} \sum_{m=-l}^l {}_{-2}Y^{lm}(\theta, \phi) h^{lm}(\lambda; t).$$

Para un par (l, m) dado, la expresión $h(\lambda, t)$ (omitendo los índices l y m) se puede caracterizar con dos polarizaciones linealmente independientes, de característica transversal, que se pueden identificar como $h_+(\lambda, t)$ y $h_\times(\lambda, t)$, que denotan dos modos de polarización, de forma que

$$h(\lambda, t) = h_+(\lambda, t) + ih_\times(\lambda, t),$$

que también se puede escribir de la forma

$$h(\lambda, t) = A(\lambda, t)e^{i\phi(\lambda, t)}.$$

También pueden interpretarse estos dos grados de libertad como polarizaciones a derecha e izquierda:

$$h_{right}(\lambda, t) = \frac{1}{\sqrt{2}}(h_p(\lambda, t) + ih_\times(\lambda, t))$$

y

$$h_{left}(\lambda, t) = \frac{1}{\sqrt{2}}(h_+(\lambda, t) - ih_\times(\lambda, t)).$$

1.2. Modelo Predictivo

El procedimiento utilizado para obtener predicciones de ondas gravitacionales está constituido por dos secciones: una *offline* y otra *online*. La etapa *offline* consta del entrenamiento del modelo reducido y la construcción del predictivo, que es donde ocurre el mayor costo computacional de todo el proceso, y la etapa *online* está constituida por la obtención de predicciones del modelo obtenido en tiempo real (en el orden de milisegundos en un computador personal).

- Etapa *offline*. Se puede dividir en tres pasos secuenciales:
 1. Del espacio de funciones en cuestión se realiza una selección de los n puntos más relevantes del espacio de parámetros. Las funciones asociadas a estos parámetros provee una base reducida cuasi-óptima para representar la familia de funciones.
 2. Se identifica n valores temporales del espacio temporal en cuestión, llamados *nodos* o *tiempos empíricos*, para construir interpolaciones (llamados interpolantes empíricos) que sean predicciones de las soluciones buscadas por el modelo. Se utiliza el Método de Interpolación Empírico (EIM) para obtener los nodos temporales y el interpolante, el cual utiliza la base reducida encontrada y el conjunto de valores temporales con los que se trabaja. Por construcción el número de valores temporales es el mismo número de funciones que construyen la base reducida.
 3. En cada tiempo empírico se realizan regresiones en el espacio de parámetros para obtener representaciones de las funciones en dichos tiempos, las cuales permiten construir interpolantes en los nodos empíricos que serán las predicciones del modelo.
- Etapa *online*. Se evalúa el modelo predictivo obtenido en cualquier parámetro dentro del espacio de parámetros que se ha construido. Esto se realiza obteniendo predicciones de las regresiones obtenidas en el espacio de parámetros, con las cuales se construye el interpolante empírico, que es la predicción otorgada por el modelo.

El enfoque utilizado para obtener el modelo predictivo es genérico. Es decir, puede utilizarse directamente para problemas involucrando series temporales parametrizadas (e incluso problemas más genéricos) de forma directa.

1.3. Enfoque del trabajo

Para el presente trabajo se construyó por completo el código para un modelo predictivo con el fin de describir el modo dominante $l = m = 2$ de las ondas gravitacionales producidas por un sistema binario de agujeros negros. El modelo consiste de un algoritmo Greedy para obtener la base reducida, un algoritmo EIM para encontrar los nodos más representativos en el tiempo, realizar regresiones en el espacio de parámetros y construir un interpolante a través de la base reducida, el cual a partir de evaluaciones de las regresiones en el espacio de parámetros otorgará representaciones de ondas gravitacionales para un parámetro especificado por el usuario.

Debido a que el factor de mayor contribución al error del modelo es la imprecisión de las regresiones, a posteriori se realiza un estudio comparativo de diferentes modelos de regresión sobre los datos obtenidos para la fase de regresión, con la finalidad de evidenciar las regresiones

de mayor utilidad para representar las ondas gravitacionales en cuestión y construir modelos predictivos lo más precisos posible.

Se compara principalmente el poder predictivo conseguido a través de una gran variedad de modelos provenientes del ámbito de Machine Learning y de métodos clásicos de aproximación, realizando regresiones globales y locales, en este último caso a través de regresiones mediante vecinos cercanos.

Para utilizar una gran variedad de modelos de Machine Learning, se utilizó la plataforma de AutoML (Machine Learning Automatizado) DataRobot [20] corriendo en el servicio de la nube Amazon Web Services (AWS), a través de varios convenios académicos gratuitos.

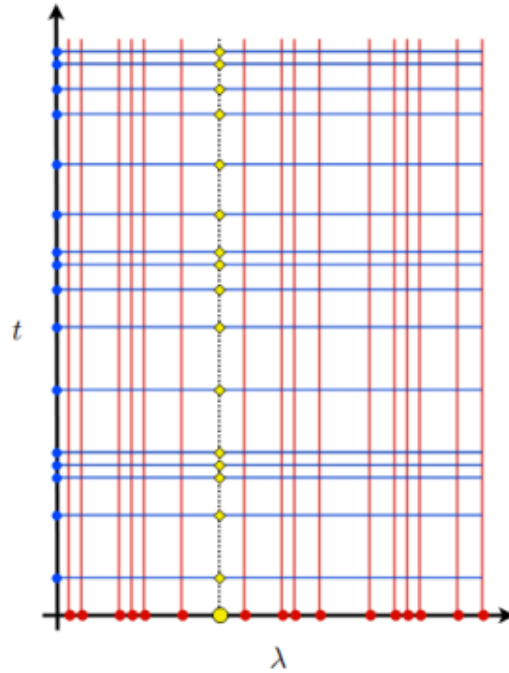


Figura 1.1: Esquema del modelo *surrogate* o sustituto. Primero se identifican parámetros Greedy, denotados con puntos rojos. Cada uno tiene asociada una solución, representadas como líneas verticales rojas. A partir de la base reducida se identifica un conjunto de nodos empíricos en el tiempo, denotados con puntos azules. Luego en cada nodo empírico se realizan regresiones para obtener una expresión de las soluciones en todo el espacio de parámetros para cada nodo, graficadas como líneas horizontales azules. Por último, para realizar una predicción para un parámetro λ dado, se evalúan las regresiones en los nodos empíricos en dicho parámetro y se obtiene un interpolante empírico, representado como una línea vertical negra, con diamantes amarillos.

Capítulo 2

Marco Teórico

En este capítulo se realiza una introducción de elementos del estado del arte de modelos de orden reducido y ciencia de ondas gravitacionales. Se dará una introducción a la teoría a tener en cuenta, como representaciones compactas, bases reducidas, interpolación polinomial, el método de interpolación empírica y la construcción de un modelo predictivo de ondas gravitacionales.

En problemas parametrizados es de interés la obtención de funciones de la forma

$$h(\lambda, \cdot) = h_\lambda(\cdot) = h_\lambda$$

para un λ dado. En particular para el presente trabajo $h(\lambda, \cdot)$ denota una expresión para ondas gravitacionales, λ denota los parámetros intrínsecos del problema que describen un par de agujeros negros, que conforman la fuente de las ondas gravitacionales de estudio, por último \cdot representa las variables extrínsecas del problema, es decir las variables espacio-temporales en cuestión.

En las áreas de computación científica y ciencia de datos es de interés realizar múltiples evaluaciones y análisis de funciones en tiempo real. En particular, en astronomía de ondas gravitacionales es de gran importancia poder realizar un número grande de predicciones de ondas gravitacionales producidas por un sistema binario de agujeros negros, principalmente con aplicación en detección de señales y estimación de parámetros.

Existen situaciones donde para obtener h_λ se necesita resolver un problema numérico de gran costo computacional, como puede ser resolver las ecuaciones de Einstein en Relatividad Numérica. Estas instancias han dado lugar a enfoques alternativos para lograr obtener soluciones a bajo costo.

Una característica de las expresiones h_λ de ondas gravitacionales es que disponen de información redundante al variar λ . Por lo tanto se puede utilizar un espacio de dimensión reducida para representar las soluciones de forma compacta.

Al realizar representaciones con diferentes espacios, en general se obtiene resultados de diferente precisión, y como consecuencia, para realizar una representación certera, se debe desarrollar una metodología para la elección del espacio. Si se elige un espacio arbitrario no se espera una precisión de representación cercana a la óptima; esto sería realizar una descripción sin pérdida de información. La elección del mejor espacio no es una tarea fácil, ya que la cantidad de espacios posibles crece combinatorialmente con la dimensión del espacio inicial, lo cual imposibilita para la mayoría de los casos realizar una búsqueda por fuerza bruta.

Como consecuencia de estas situaciones surge con naturalidad la pregunta de que tan bien uno puede realizar una representación de h_λ ($\forall \lambda$ en el intervalo de interés) a partir de un espacio

dado. La métrica de Kolmogorov ayuda a cuantificar esto, que será introducida en el presente texto.

Un enfoque posible al problema es el de intentar buscar un conjunto minimal y cuasi-óptimo de soluciones $\{h_{\Lambda_i}\}_{i=1}^n$ que represente el conjunto h_λ . El enfoque de bases reducidas junto a un algoritmo Greedy ayuda a encontrarlo.

Para obtener expresiones de h_λ , se puede pensar en obtener expresiones a través del método de proyección, que es óptimo para el problema de cuadrados mínimos, pero ocurre que para obtener una proyección se necesita conocer por completo la función a proyectar, de forma que no se ahorra costo computacional. Un método alternativo a la proyección es la interpolación, que requiere de menos información para ser calculada.

Se puede elegir los nodos más convenientes para interpolar a través de $\{h_{\Lambda_i}\}_{i=1}^n$, utilizando el método de interpolación empírica (EIM). Este método es altamente efectivo, debido a que se centra en minimizar la constante de Lebesgue para obtener un interpolante de alto poder predictivo.

Si $\{T_i\}_{i=1}^n$ son los nodos obtenidos con el método EIM, luego se pueden realizar regresiones en λ para cada T_i para obtener $h(\lambda, T_i) \forall \lambda$. De esta forma se tiene toda la información necesaria para conformar un modelo predictivo. A continuación se presenta una introducción teórica para la construcción del mismo.

2.1. Problema de cuadrados mínimos (Least Squares)

A menudo es útil aproximar un vector con un espacio conformado por otros vectores. El rango de aplicación es grande: cómputo de integrales, resolución de ecuaciones diferenciales y compresión de imágenes, por nombrar algunos temas. En cuanto a funciones el desarrollo de Taylor constituye un ejemplo conocido, donde el espacio aproximante está generado por $\{(x-a)^n | a \in \mathbb{R}\}_{n=0}^N$, con N dependiente del problema en cuestión y generalmente otorgando mayor precisión a mayor N .

Considerando un espacio de Hilbert¹ \mathcal{H} y un subespacio vectorial $W_n \subset \mathcal{H}$ n -dimensional, un criterio común de aproximación de un $v \in \mathcal{H}$ por W_n es el de cuadrados mínimos (LS, por su nombre en inglés). La mejor representación de v a partir de un $w \in W_n$ estará dada por el w que cumpla la condición

$$\begin{aligned} w &= \underset{W_n}{\operatorname{argmin}} \|v - w\|^2 \\ &= \underset{W_n}{\operatorname{argmin}} v \cdot w . \end{aligned}$$

La solución w es la proyección ortogonal² de v en W_n , es decir $w = \mathcal{P}_{W_n} v$ con $\mathcal{P}_{W_n} = \sum_{i=1}^n |e_i\rangle \langle e_i|$, donde $\{e_i\}_{i=1}^n$ es una base ortonormal de W_n .

Por lo tanto

$$w = \sum_{i=1}^n (e_i \cdot v) e_i ,$$

donde \cdot denota el producto interno entre dos vectores.

Esto significa que si $v \in W_n$

$$\mathcal{P}_{W_n} v = v ,$$

¹Espacio de Hilbert: espacio vectorial normado $(V, \|\cdot\|)$ y completo (toda sucesión de Cauchy con respecto a su norma tiene un límite en V), cuya norma deviene de un producto escalar $V \times V \rightarrow \mathbb{C}$. Es decir, un espacio de Banach con norma proveniente de un producto escalar.

²Un operador P es un proyector ortogonal si cumple $P = P^2$ y $P = P^\dagger$, donde P^\dagger representa al adjunto de P , es decir, su transpuesto conjugado.

y el residual $\delta v = \mathcal{P}_{W_n} v - v$ satisface

$$(\mathcal{P}_{W_n} v) \cdot \delta v = 0 .$$

En cuanto a cuestiones computacionales es preferible utilizar bases ortonormales debido a problemas de condicionamiento. Se desarrollará este asunto más adelante.

2.2. Representación de un conjunto de elementos: error colectivo

Si se busca aproximar un conjunto de elementos $\mathcal{F} = \{v_1, \dots, v_n\}$ a partir de un subespacio W_m (donde $\dim(W_m) = m$), se necesita cuantificar la precisión de representación a través de un criterio dado. La elección de la métrica que represente el error colectivo en general depende del problema en cuestión. Dos definiciones razonables para la métrica son el error medio

$$\frac{1}{N} \sum_{i=1}^n \|v_i - \mathcal{P}v_i\| ,$$

que resulta ser la media aritmética del cuadrado del conjunto de errores, por lo que muestra un promedio y no da indicios sobre errores máximos o mínimos dentro del conjunto representado, y el error máximo o norma infinito

$$\max_i \|v_i - \mathcal{P}v_i\| .$$

que posee mayor estrictez que el primero, al solo tomar en cuenta el peor elemento representado.

Luego de saber como cuantificar la precisión de una representación, uno se puede preguntar como para un dado \mathcal{F} se puede encontrar un subespacio W_m que permita una buena aproximación de este, lo cual es no trivial y se analizará más adelante.

2.3. Métodos espectrales

En esta sección se discute la rápida convergencia de las aproximaciones lineales clásicas, que naturalmente conducen a Bases Reducidas.

Supóngase considerar un conjunto de funciones complejas \mathcal{F} . La aproximación lineal estándar consiste en una base de n elementos $\{e_i(x)\}_{i=1}^n$ elegida para dar lugar a una aproximación de una función $h(x)$ en \mathcal{F} expresada como una combinación lineal de los elementos de la base

$$h(x) \approx \sum_{i=1}^n c_i e_i(x) ,$$

donde los coeficientes de expansión c_i se pueden elegir según diferentes criterios, por ejemplo a través del problema de LS y a través del problema de interpolación, que se tratará más adelante.

En cuanto al problema de LS, los coeficientes c_i son elegidos de tal forma que minimicen el error de representación

$$\left\| h(\cdot) - \sum_{i=1}^n c_i e_i(\cdot) \right\|^2 .$$

Como ya se sabe, la solución al problema de LS es la proyección ortogonal con respecto al span de la base, que cumple

$$\mathcal{P}_{W_n} h = \sum_{i=1}^n c_i e_i \iff \{c_i\}_{i=1}^n = \operatorname{argmin}_{\{c'_i\}_{i=1}^n} \left\| h - \sum_{i=1}^n c'_i e_i \right\|^2$$

con

$$c_i = e_i \cdot h$$

si $\{e_i\}_{i=1}^n$ es una base ortonormal.

La tasa de convergencia es un término referido a la tasa de decaimiento del error para un h dado a medida que n crece:

$$\|h - \mathcal{P}_{W_n} h\| .$$

En términos de precisión y convergencia para aproximar un espacio \mathcal{F} , todavía no se discutió un aspecto que será de importancia, que es la elección de un W_n con el cual se obtenga representaciones precisas.

Uno de los casos más familiares de métodos espectrales es el de expansiones de Fourier, donde se utiliza funciones periódicas $e_j = e^{ijx}/\sqrt{2\pi}$ con $j \in \mathbb{Z}$ y $x \in [0, 2\pi]$.

Si por ejemplo se utiliza n par por simplicidad, se obtiene

$$\mathcal{P}_{W_{n+1}} h(x) = \sum_{i=-n/2}^{n/2} c_i e_i(x) = \sum_{i=-n/2}^{n/2} (e_i \cdot h) e_i(x) ,$$

con $W_{n+1} = \operatorname{span}\{e_{-n/2}, e_{(-n+1)/2}, \dots, e_{n/2}\}$. Si h tiene definidas las s primeras derivadas, entonces existe $C \in \mathbb{R}_{>0}$ tal que la tasa de convergencia está dada por:

$$\|h - \mathcal{P}_{W_{n+1}} h\| \leq C(n+1)^{-s} \left\| \frac{\partial^s h}{\partial x^s} \right\| , \quad \forall n \geq 0 .$$

En particular si h es una función C^∞ , es decir, el caso anterior pero con $s \rightarrow \infty$, la tasa de convergencia decae más rápido que cualquier ley de potencia. Esta cualidad se denomina convergencia espectral.

A partir de esto se puede concluir que la expansión en modos de Fourier provee una rápida convergencia, que es más acentuada mientras más *suave* es la función a representar. El inconveniente que pueden ocasionar estos modos es que son funciones periódicas, y pueden no ser óptimos para representar ciertos comportamientos, por ejemplo no periódicos.

Un caso de bases no periódicas son las construidas a través de polinomios e el intervalo $[-1, 1]$. Para obtener una, se puede recurrir al conjunto de polinomios $\{x^i\}_{i=0}^n$ y aplicar a este el procedimiento de Gram-Schmidt. La base obtenida será el conjunto de los primeros $n+1$ Polinomios de Legendre. La fórmula de Rodrigues los expresa como $P_n = \frac{1}{2^n n!} \frac{\partial^n}{\partial x^n} (x^2 - 1)^n$. A partir de esta base se obtiene la tasa de convergencia:

$$\|h - \mathcal{P}_{W_{n+1}} h\| \leq C(n+1)^{-s} \left\| (1-x^2)^{s/2} \frac{\partial^s h}{\partial x^s} \right\| \leq C(n+1)^{-s} \left\| \frac{\partial^s h}{\partial x^s} \right\| , \quad \forall n > s-1 ,$$

con $W_{n+1} = \operatorname{span}\{P_0, P_1, \dots, P_n\}$, y además

$$\|h - \mathcal{P}_{W_{n+1}} h\| < C(n+1)^{-s} \left\| \frac{\partial^s h}{\partial x^s} \right\| , \quad \forall n > s-1 \text{ y } x \neq 0 ,$$

donde \leq se cumple solo para $x = 0$, por lo que se puede esperar que a través de una base polinomial se obtenga mejor convergencia que con una base de Fourier.

Si h es C^∞ para este caso también se obtiene convergencia espectral.

2.4. Distancia de Kolmogorov

Sea \mathcal{K} un espacio de funciones de interés y $W_{\lambda',n}$ un espacio de Hilbert n-dimensional parametrizado con un parámetro λ' (puede ser multidimensional). Para cada función $h \in \mathcal{K}$, se puede definir la distancia entre el subespacio $W_{\lambda',n}$ y la función h . Eligiendo una base ortonormal de $W_{\lambda',n}$, denotada como $\{e_i\}_{i=1}^n$, esto se puede escribir de la forma

$$\mathcal{D}_n(h, W_{\lambda',n}) = \min_{c_i} \left\| h - \sum_{i=1}^n c_i e_i \right\|^2.$$

Usualmente el interés no se centra en la distancia de $W_{\lambda',n}$ con un único $h \in \mathcal{K}$, sino en qué tan bien se aproxima todo el conjunto de funciones de \mathcal{K} . En el caso que todo $h \in \mathcal{F}$ esté parametrizado con un parámetro λ , se tiene $h = h(\lambda)$ y se puede definir

$$\mathcal{D}_n(\mathcal{F}, W_{\lambda',n}) = \max_{\lambda} \min_{c_i} \left\| h(\lambda) - \sum_{i=1}^n c_i(\lambda) e_i \right\|^2,$$

que permite conocer el error de representación de $W_{\lambda',n}$ sobre el elemento de \mathcal{F} que *peor* logra describir, es decir, esta es una cota superior del error de aproximación sobre \mathcal{F} .

Dados dos subespacios n-dimensionales W_1 y W_2 , la razón

$$\kappa(\mathcal{F}, W_1, W_2; \mathcal{K}) = \frac{\mathcal{D}_n(\mathcal{F}, W_1; \mathcal{F})}{\mathcal{D}_n(\mathcal{F}, W_2; \mathcal{F})}$$

permite obtener la capacidad de aproximación de W_1 sobre \mathcal{F} comparada a la de W_2 . Si $\kappa \approx 1$, a priori no se tiene una razón para preferir un espacio sobre el otro. Si $\kappa \ll 1$ se prefiere W_1 y si $\kappa \gg 1$, se prefiere W_2 . Este criterio es útil para comparar las propiedades de aproximación de dos espacios, pero usualmente es de mayor interés la comparación entre múltiples espacios. Una pregunta natural es la comparación de un subespacio W_i con todos los otros subespacios n-dimensionales de \mathcal{K} . El mayor interés reside en encontrar el mejor espacio que aproxime a \mathcal{F} (¿Cuál otro sino?). Este será el que cumpla

$$\begin{aligned} d_n(\mathcal{F}, W_{\lambda',n}) &:= \min_{W_{\lambda',n}} \mathcal{D}_n(\mathcal{F}, W_{\lambda',n}) \\ &= \min_{W_{\lambda',n}} \max_{\lambda} \min_{c_i} \left\| h(\lambda, \cdot) - \sum_{i=1}^n c_i(\lambda) e_i(\cdot) \right\|^2. \end{aligned}$$

A partir de la solución al problema de LS, se tiene

$$d_n(\mathcal{F}, W_{\lambda',n}) = \min_{W_{\lambda',n}} \max_{\lambda} \|h(\lambda, \cdot) - \mathcal{P}_{W_{\lambda',n}} h(\lambda, \cdot)\|^2,$$

donde $\mathcal{P}_{W_{\lambda',n}}$ es la proyección ortogonal sobre $W_{\lambda',n}$.

Por lo tanto, un subespacio W_i es óptimo si

$$\min_{W_{\lambda',n}} \max_{\lambda} \|h(\lambda, \cdot) - \mathcal{P}_n h(\lambda, \cdot)\|^2 = \max_{\lambda} \|h(\lambda, \cdot) - \mathcal{P}_{W_i} h(\lambda, \cdot)\|^2.$$

$d_n(\mathcal{F}, W_{\lambda', n})$ es conocida como la distancia de Kolmogorov, que define la cota superior de aproximación a través de la mejor solución de LS, utilizando el mejor subespacio para aproximar $W_{\lambda, n}$.

Si se analiza la medida de Kolmogorov en función de n , se la puede interpretar como la tasa de convergencia de la cota superior del error de aproximación, donde para cada n se elige el mejor subespacio de aproximación posible.

Esta tasa de convergencia depende de hasta que derivada s sea continuo el mapa $\lambda \rightarrow h(\lambda, \cdot)$, al igual que en expansiones espectrales. Si $\lambda \rightarrow h(\lambda, \cdot) \in C^\infty$ se puede esperar convergencia más rápida que cualquier ley de potencia.

Se puede observar que la distancia de Kolmogorov se mide a través del error máximo en el espacio de parámetros, denotado por λ . Una variante a esta puede ser definida si en vez de tomar el error máximo en λ , se utiliza el error medio (a lo largo del volumen del espacio de parámetros λ). De esta forma se obtendría la distancia

$$d_n(\mathcal{F}_\lambda, W_{\lambda', n}) = \min_{W_{\lambda', n}} \frac{\int_\lambda \|h(\lambda, \cdot) - \mathcal{P}_n h(\lambda, \cdot)\|^2 d\lambda}{\int_\lambda d\lambda}.$$

Con estos resultados se puede clarificar el problema del mejor subespacio aproximante para un espacio \mathcal{F} , ya que se conoce la solución al problema de LS y se dispone de una distancia para cuantificar que tan buena es una representación, y en caso de encontrar el espacio que la minimiza, se tiene la mejor representación posible.

La introducción de la medida de Kolmogorov es esclarecedora, pero se debe fijar atención en un aspecto: en general existe gran dificultad de resolución del problema de forma óptima. Para hacerlo se debe comparar la capacidad de aproximación de todos los subespacios formados con diferentes combinaciones de elementos de la base del espacio. Esto conlleva un problema de complejidad combinatorial en términos computacionales, lo cual implica un costo grande para su resolución en un lapso de tiempo razonable.

Esto nos sugiere afrontar el problema de Kolmogorov con un método alternativo al de comparar la precisión de todos los subespacios posibles. Como se verá más adelante, existe un enfoque que permite obtener una solución cuasi-óptima, que es el de Bases Reducidas.

2.5. Bases Reducidas

El enfoque de Bases Reducidas (RB, por su nombre en inglés) es de utilidad para encontrar un conjunto reducido de funciones que sean representativas de un conjunto dado de soluciones, específicas para un problema en cuestión.

Se busca un conjunto

$$\{h_{\Lambda_i}\}_{i=1}^n$$

con parámetros Λ_i elegidos cuidadosamente del espacio de parámetros λ en cuestión, donde $\dim(\{h_{\Lambda_i}\}_{i=1}^n) = n$. El conjunto de soluciones $\{h_{\Lambda_i}\}_{i=1}^n$ (o una combinación lineal de ellas) constituyen la base reducida

Se identifica a

$$\text{span}(\{h_{\Lambda_i}\}_{i=1}^n)$$

como el espacio correspondiente a la base reducida.

Por razones de condicionamiento numérico, puede ser conveniente utilizar con un conjunto ortonormalizado $\{e_i\}_{i=1}^n$ en lugar de $\{h_{\Lambda_i}\}_{i=1}^n$, que puede ser obtenido a través del método de Gram-Schmidt.

La aproximación RB para una solución h_λ dada es $\mathcal{P}_n h_\lambda$, donde \mathcal{P}_n denota la proyección ortogonal sobre el espacio $\text{span}(\{h_{\Lambda_i}\}_{i=1}^n)$.

Si el conjunto de funciones de interés h_λ tiene dependencia suave con respecto a los parámetros λ , se espera que la expansión tenga convergencia muy rápida a medida que se aumenta el número de bases n . Esto es similar a la rápida convergencia obtenida para métodos espectrales. En el caso de RB se explota la suavidad en los parámetros, ya que los elementos de la base reducida son tomados del espacio de funciones de interés.

Más adelante se verá que además de construir representaciones de funciones se puede construir predicciones para cualquier parámetro λ dentro del intervalo de interés. Los métodos utilizados con este fin llevan el nombre de modelos predictivos, ya que para construir cualquier h_λ no necesitan su representación fidedigna. Existen diversos enfoques para obtener un modelo predictivo, por ejemplo una forma es aproximar los coeficientes de proyección de las funciones de la base reducida para todo el espacio de parámetros, y otra es a través del algoritmo E.I.M., el cual se introducirá más adelante.

Un método efectivo para encontrar una base reducida es a través de un algoritmo Greedy que se enfoca en encontrar las funciones más disímiles dentro de una discretización del espacio de parámetros. El algoritmo Greedy utilizado tiene una convergencia muy rápida y resuelve de forma casi-óptima el problema de Kolmogorov, que consiste en encontrar el subespacio que da el menor error de aproximación para una dimensión dada.

2.6. Algoritmos Greedy o Voraces

De forma genérica, los algoritmos Greedy (o voraces, en español) son un paradigma que construye soluciones a un problema dado, armándolas pieza a pieza, siempre eligiendo en cada paso la que ofrece el más inmediato beneficio. Los problemas donde elegir soluciones localmente óptimas derivan en soluciones globales son indicados para este enfoque. Normalmente se trata de algoritmos que resuelven problemas de optimización.

Ingredientes comunes de los algoritmos voraces: se tiene un problema a resolver de manera óptima, un conjunto de candidatos a integrar la solución y una métrica para saber si una solución parcial satisface la condición mínima de optimalidad especificada por el usuario. Cuando se cumple dicha condición se detiene el algoritmo, indicando que ha convergido a una solución del problema.

Procedimiento general de los algoritmos Greedy: Puede que inicialmente ningún candidato haya sido considerado o que se elija a uno como semilla del algoritmo. Esto depende del problema en cuestión. Luego se realiza un loop, donde en cada iteración se utiliza una función de selección para elegir un integrante de la solución final, que será el mejor candidato para las condiciones dadas en la iteración (de aquí proviene el nombre "voraz"). El loop finaliza en caso de que se cumpla la condición mínima de optimalidad.

Algorithm 1 Algoritmo Greedy

- 1: $C = \{h_i\}_{i=1, \dots, M}$ (Candidates)
 - 2: $S = \{ \}$ (Solution)
 - 3: **while** S not solution **do**
 - 4: Select best candidate h_i of C
 - 5: $S = S \cup \{h_i\}$
 - 6: **Output:** S
-

Los algoritmos Greedy nunca revisan una elección ya realizada, en ellos se asume haber

elegido de forma acertada las componentes anteriores. Por ello, no todos los problemas admiten solución voraz, pero muchos problemas de interés sí admiten solución. Un ejemplo es el problema del camino de costo mínimo en un grafo (problema de Dijkstra), que se puede interpretar como la búsqueda de una geodésica en un grafo.

En muchos problemas, una estrategia Greedy no suele producir una solución óptima, pero, no obstante, una heurística de este tipo puede producir soluciones localmente óptimas que se aproximen a una solución globalmente óptima en un período de tiempo razonable.

2.7. Algoritmo Greedy para obtener una Base Reducida

Un método utilizado para obtener una base reducida es a través de un algoritmo Greedy. Se comienza con un conjunto representativo de soluciones $\mathcal{T} = \{h_{\lambda_i}\}_{i=1}^N$ (llamado espacio de entrenamiento), que es una discretización del espacio de soluciones. El algoritmo identifica un conjunto de parámetros $\{\Lambda_i\}_{i=1}^n \subset \{\lambda_i\}_{i=1}^N$ tal que sus funciones asociadas $\{h_{\Lambda}\}_{i=1}^n$ constituyen un conjunto representativo de las funciones del espacio de entrenamiento, que es minimal y cuasi-óptimo.

La idea principal es construir una base reducida tal que

$$\|h(\lambda, \cdot) - P_n h(\lambda, \cdot)\|^2 \leq \epsilon \quad \forall \lambda \in \mathcal{T},$$

donde P_n representa a la proyección ortogonal sobre el espacio generado por la base reducida.

El espacio de entrenamiento \mathcal{T} consiste de un conjunto de soluciones obtenidas a través de una discretización del espacio de parámetros. Puede ser construido a través de un muestreo uniforme o aleatorio. Sin importar el método de construcción, el formalismo RB provee una representación precisa de \mathcal{T} .

Denotando a P_i como la proyección ortogonal sobre la base reducida con i elementos, el algoritmo Greedy para encontrar los h_{λ_i} más representativos de \mathcal{T} es el siguiente:

Algorithm 2 Algoritmo Greedy para encontrar una Base Reducida

- 1: **Input:** $\mathcal{T} = \{\lambda_i, h(\lambda_i, x)\}_{i=1}^N, \epsilon$
 - 2: Set $i=1$
 - 3: **Seed choice** (arbitrary): $\Lambda_i \in \{\lambda_j\}_{j=1}^N$
 - 4: $rb = \{h(\cdot, \lambda_i) / \|h(\cdot, \lambda_i)\|\}$
 - 5: $\sigma_i = \max_{\lambda \in \mathcal{T}} \|h(\cdot, \lambda) - P_i h(\cdot, \lambda)\|^2$
 - 6: **while** $\sigma_i > \epsilon$ **do**
 - 7: $i=i+1$
 - 8: $\Lambda_i = \operatorname{argmax}_{\lambda \in \mathcal{T}} \|h(\cdot, \lambda) - P_{i-1} h(\cdot, \lambda)\|^2$
 - 9: $e_i = h(\cdot, \Lambda_i) - P_{i-1} h(\cdot, \Lambda_i)$ (Gram-Schmidt)
 - 10: $e_i = e_i / \|e_i\|$ (normalización)
 - 11: $rb = rb \cup e_i$
 - 12: $\sigma_i = \max_{\lambda \in \mathcal{T}} \|h(\cdot, \lambda) - P_i h(\cdot, \lambda)\|^2$
 - 13: **Output:** $rb = \{e_i\}_{i=1}^n$ y puntos Greedy $\{\Lambda_i\}_{i=1}^n$
-

Se comienza con un error inicial $\sigma_0 = \max_{\lambda \in \mathcal{T}} \|h(\lambda, \cdot) - P_0 h(\lambda, \cdot)\|^2$, donde \mathcal{P}_0 indica la proyección sobre el espacio generado por la función elegida como semilla. En caso de que $\sigma_0 < \epsilon$, se habrá obtenido la solución deseada y el algoritmo se detendrá, de lo contrario se continuará buscando la solución, ingresando en el loop while del algoritmo hasta converger a una solución.

Cada vez que se elige un nuevo elemento para la base reducida, en el algoritmo se calcula la capacidad de aproximación de la nueva base. En caso de no cumplirse la condición mínima de optimalidad, se procederá a enriquecer la base con un nuevo elemento del espacio de entrenamiento las veces que sea necesario hasta cumplir con esta condición.

Detalles del algoritmo:

- En la línea 8 lo que se hace es seleccionar el elemento del espacio de entrenamiento que **peor** se ha representado con la RB construida hasta la iteración dada, para luego agregarla a la base. Esta elección es la que caracteriza al algoritmo como Greedy, la que permite encontrar representación cuasi-óptima de \mathcal{T} y una solución cuasi-óptima del problema de Kolmogorov. Más adelante se tratará el algoritmo EIM que tiene un enfoque muy similar.
- Notar que el cálculo del error de aproximación de cada elemento del espacio de entrenamiento para una base dada es independiente de los restantes (línea 8 del algoritmo), por lo cual el cálculo llevado a cabo en esta línea es altamente paralelizable, en sentido computacional.
- Por construcción, la complejidad computacional de aumentar la base en un elemento es independiente de la dimensionalidad de ella. Esto se debe a que en cada iteración se guarda la información de las proyecciones calculadas sobre cada elemento del espacio de entrenamiento, que implica que en cada paso se tenga que calcular solo un nuevo coeficiente de proyección por elemento del espacio de entrenamiento. Por lo tanto el costo de construir una RB con n elementos es lineal en n ($\mathcal{O}(n)$).
- Gracias a ser un algoritmo de optimización global, la elección de la semilla es irrelevante para el problema en cuestión.

- Los parámetros Greedy $\{\Lambda_i\}_{i=1}^n$ son jerárquicos, es decir, que para mayor (menor) precisión se debe agregar (quitar) puntos del conjunto conseguido, debido a la buena convergencia del algoritmo. Se tratará la convergencia del algoritmo más adelante.
- En caso que haya una parcial redundancia entre las funciones de \mathcal{T} se obtendrá $n < N$ o eventualmente $n \ll N$.
- Se trabaja con una base ortonormal para evitar el mal condicionamiento al momento de computar una proyección. Esto no soluciona totalmente el problema, ya que para evitar la acumulación de error numérico en el proceso de Gram-Schmidt se debe utilizar una variación de este, llamado Gram-Schmidt Iterado.

Otro error de interés es el asociado a la representación de cualquier función h_λ , no necesariamente en el conjunto de entrenamiento, dado por

$$\delta(\lambda) := \|h(\lambda, \cdot) - P_n h(\lambda, \cdot)\|^2 .$$

Este error representa una validación de la base reducida sobre todo el espacio de parámetros. Se puede estimar realizando un sampleo aleatorio de λ y computando los errores asociados a cada parámetro.

2.7.1. Convergencia del Algoritmo Greedy

Primero se puede notar que la elección aleatoria de la semilla supone un costo en la representatividad de la base reducida, ya que elegir un elemento de forma aleatoria supone una metodología subóptima. Pero a pesar de esto se puede mostrar que el algoritmo tiene muy buena convergencia [13].

Se puede saber de la cuasi-optimalidad del algoritmo Greedy a través de la distancia de Kolmogorov d_n [8]. Si ϵ es la tolerancia dada por el usuario para el algoritmo Greedy, en caso de que haya convergencia se tendrá

$$\sigma_n := \max_{\lambda \in \{\Lambda_i\}_{i=1}^n} \|h(\lambda, \cdot) - P_{rb} h(\lambda, \cdot)\|^2 \leq \epsilon .$$

σ_n es designado como el *error Greedy*, que cuantifica el error máximo de la representación de \mathcal{T} obtenido a través de la base reducida. Notar que esta distancia también se puede definir sobre todo el espacio de parámetros, extendiendo $\max_{\lambda \in \{\Lambda_i\}_{i=1}^n}$ a $\max_{\lambda \in \mathcal{D}}$, siendo \mathcal{D} el espacio de parámetros completo.

Si se define a \mathcal{G} como la función de búsqueda de una base reducida a través del algoritmo Greedy, se puede observar cierta similitud entre d_n y σ_n :

$$d_n(\mathcal{F}, W_{\lambda,n}) = \min_{W_{\lambda,n}} \left(\max_{\lambda} \|h(\lambda, \cdot) - \mathcal{P}_n h(\lambda, \cdot)\|^2 \right)$$

$$\sigma_n(\mathcal{T}, rb) = \mathcal{G} \left(\max_{\lambda} \|h(\lambda, \cdot) - \mathcal{P}_n h(\lambda, \cdot)\|^2 \right)$$

donde \mathcal{F} es el espacio de soluciones del problema, $W_{\lambda,n} \subset \mathcal{H}$ es el mejor subespacio para d_n , \mathcal{T} es el espacio de entrenamiento y $rb = \text{span}\{h_\Lambda\}_{i=1}^n$. Uno quisiera ver que tan comparables son los resultados para d_n y σ_n , ya que d_n es una distancia con respecto al subespacio óptimo de aproximación.

La convergencia de d_n depende de la suavidad en λ de las soluciones $h_\lambda \in \mathcal{F}$. Si las funciones son analíticas se puede probar que d_n decae exponencialmente con n .

Además, un resultado para d_n y σ_n es el siguiente:

$$d_n \leq De^{-an^b} \implies \sigma_n \leq \sqrt{2D}e^{-\alpha n^b}, \quad \alpha = a2^{-1-2b} \text{ y } D, a, b > 0.$$

En los casos de interés como se cumple que $h_\lambda \in \mathcal{F}$ tienen una dependencia suave con λ , se puede afirmar que el error del algoritmo Greedy σ_n decae exponencialmente con n .

Por lo tanto la expansión RB tiene una convergencia rápida al espacio \mathcal{F} a medida que se agregan elementos a la base reducida a través del algoritmo Greedy.

2.8. Interpolación polinomial

El método de interpolación polinomial es de interés ya que tiene una relación cercana a métodos espectrales, esto se puede ver por ejemplo a través de la constante de Lebesgue, la cual se introducirá a continuación. Además este método es de interés ya que es utilizado en modelos de orden reducido aplicados a ondas gravitacionales.

Si se tiene una cantidad acotada de información de una función $f(x)$, dada por $\{f_i\}_{i=0}^n$ en nodos $\{x_i\}_{i=0}^n$ (llamados nodos interpolantes) se puede realizar una reconstrucción de esta a partir de una función $\mathcal{I}_n f$ que cumpla

$$\mathcal{I}_n f(x_i) = f_i \quad \forall i = 0, \dots, n,$$

construida a través de una base de funciones $\{e_i(x)\}_{i=0}^n$, dada por

$$\mathcal{I}_n f(x) = \sum_{i=0}^n C_i e_i(x).$$

Para el problema de LS se sabe que la solución óptima está dada por la proyección ortogonal de la función sobre la base $\{e_i(x)\}_{i=0}^n$, por lo tanto como en principio $\{x_i\}_{i=0}^n$ es un conjunto arbitrario de nodos, la solución obtenida será sub-óptima para el problema de LS, ya que para realizar una proyección se necesita conocer $f(x)$ de forma íntegra. El beneficio que provee una interpolación es la capacidad de construir una aproximación de $f(x)$ con una cantidad acotada de información. A partir de $\mathcal{I}_n f(x_i) = f_i$. Se obtiene el sistema de ecuaciones $(n+1) \times (n+1)$

$$\mathcal{I}_n f(x_j) = \sum_{i=0}^n V_{ji} C_i = f(x_j),$$

donde V_{ji} es la matriz de Vandermonde, dada por $V_{ji} = e_i(x_j)$. Por lo tanto

$$C_i = \sum_{j=0}^n (V^{-1})_{ij} f(x_j).$$

Para estudiar la convergencia es necesario introducir el error de interpolación

$$E_n(x) := |f(x) - \mathcal{I}_n f(x)|.$$

Se puede probar que

$$E_n(x) = \left| \frac{f^{(n+1)}(\eta)}{(n+1)!} w_{n+1}(x) \right|, \quad \eta \in [x_0, x_n],$$

donde

$$w_{n+1} = \prod_{i=0}^n (x - x_i).$$

Se puede pensar que a medida que se realiza una interpolación con mayor cantidad de nodos de una función $E_n \rightarrow 0$, pero existe un contraejemplo para esto, dado por el fenómeno Runge [26], que consiste que al interpolar $f(x) = \frac{1}{1+(5x)^2}, x \in [-1, 1]$ con n nodos equiespaciados, al hacer $n \rightarrow \infty, E_n \rightarrow \infty$ en los bordes del intervalo. El problema reside en la colocación de los nodos.

Si se analiza $|w_{n+1}(x)|$ se puede probar que

$$\max_{x \in [-1, 1]} |w_{n+1}(x)| \geq 2^{-n} ,$$

donde en particular para los nodos de Chebyshev se cumple $\max_{x \in [-1, 1]} |w_{n+1}(x)| = 2^{-n}$, esto quiere decir que con ellos se minimiza el error proveniente de la colocación de los nodos.

2.8.1. Splines

Una de las desventajas de la interpolación polinomial son las grandes oscilaciones producidas para grados de polinomios suficientemente altos (el fenómeno Runge es una evidencia de esto).

Para mantener la simplicidad de las aproximaciones con polinomios y evitar el fenómeno de Runge, se puede recurrir a aproximar una curva a través de una curva construida por partes con polinomios.

Una curva Spline S de grado k cumple:

- Está definida en un intervalo $[a, b]$
- $S \in C^{k-1}$
- Existen puntos $a = x_0, \dots, x_n = b$ llamados nudos o *knots*, donde S es un polinomio en cada subintervalo $[x_i, x_{i+1}]$.
- La curva obtenida es un interpolante de los nudos.

El método de Splines más utilizado es el Cúbico, construido a partir de polinomios de grado 3. Esto tiene varias motivaciones, como la evidente simplicidad de la curva y la capacidad de adaptarse a diferentes comportamientos que puedan tener los datos, ya que este tipo de polinomios admite inflexiones en su comportamiento (este grado de polinomios es el menor que permite esta característica).

Las curvas obtenidas a través de Splines Cúbicos son C^2 . Por lo tanto se tienen $4n$ parámetros a definir, correspondientes a los coeficientes de los polinomios (4 por cada polinomio). Se impone continuidad hasta la segunda derivada, que implica que habrá $3(n - 1)$ condiciones para las derivadas C^0, C^1, C^2 y $n + 1$ condiciones de interpolación, con un total de $4n - 2$ condiciones.

Como se puede ver, de esta forma quedan 2 coeficientes para definir la curva de forma unívoca. Una opción para tratar esta situación es utilizar la condición *not-a-knot*, donde no se utiliza como nodos al segundo y penúltimo nodos de interpolación, utilizando un único polinomio para los primeros tres puntos, y lo mismo para los últimos 3. De esta forma se tienen menos coeficientes por definir: $n - 2$ polinomios con $4(n - 2)$ coeficientes. Por otro lado hay $n + 1$ condiciones de interpolación con $3(n - 3)$ condiciones para las derivadas. Por lo tanto el numero de condiciones es el mismo numero que de coeficientes a determinar.

En cuanto a la convergencia de Splines, por ejemplo si se considera un conjunto de $n + 1$ puntos equiespaciados $x_j = x_0 + j\Delta x, j = 0, \dots, n$, el error del interpolante en cada subintervalo $[x_j, x_{j+1}]$ está acotado por

$$E_m(x) = \frac{1}{(m + 1)!} \max_{x \in [x_j, x_{j+1}]} |f^{(m+1)}(x)| \Delta x^m .$$

Como m en el caso de Splines está fijo, si $\Delta x \rightarrow 0$, $E_m(x) \rightarrow 0$. Notar que el caso de nodos equiespaciados se puede generalizar a una distribución arbitraria. Por lo tanto queda en evidencia la ausencia del fenómeno Runge.

2.8.2. Convergencia: Constante de Lebesgue

Como ya se discutió anteriormente, para el problema de LS

$$\|f - \mathcal{I}_n f\|^2 \geq \|f - \mathcal{P}_n f\|^2$$

es natural preguntarse que tan sub-óptima es una representación obtenida a través de interpolación. La constante de Lebesgue, dada por

$$\Lambda_n := \|\mathcal{I}_n\|^2 = \max_{\|h\|=1} \|\mathcal{I}_n[h]\|^2,$$

es una cota superior para la incerteza de una aproximación por interpolación, ya que se puede probar que:

$$\|f - \mathcal{I}_n f\|^2 \leq \Lambda_n \|f - \mathcal{P}_n f\|^2.$$

Como se puede ver, Λ_n establece un vínculo entre la proyección y la interpolación.

En términos de precisión, lo ideal es obtener un interpolante tal que la constante de Lebesgue crezca lo menos posible a medida que n aumenta. Computando esta constante, se puede tener una noción de la pérdida de poder predictivo de una interpolación con respecto a la mejor representación del problema de LS, es decir, la proyección.

En este caso se define a la constante de Lebesgue a través de la norma L_2 con la finalidad de trabajar siempre con la misma medida, aunque lo usual es definirla a través de la norma L_∞ ³.

Para los casos en que la constante de Lebesgue se define a través de la norma infinito, se puede ver que para interpolación polinomial con nodos equiespaciados $\Lambda_n \approx \frac{2^{n+1}}{n \log(n)}$ y para nodos de Chebyshev (óptimos para interpolación polinomial) $\Lambda_n \approx \log(n)$.

2.8.3. Representación vs. predicción

Es necesario notar una serie de aspectos con respecto a proyección e interpolación:

- Si se quiere calcular los coeficientes de proyección para una función dada de forma exacta, se necesita conocer la función a aproximar en todo su dominio, y esta información no siempre está a disposición.
- La proyección es una herramienta de representación y no de predicción, al contrario de interpolación, que estima información inicialmente desconocida.
- Al realizar una interpolación en vez de una proyección se resigna precisión, pero a la vez, se necesita menor cantidad de información para obtener una representación.
- Si no se conoce una función de forma completa, no se puede obtener una representación por proyección, por lo tanto si se quiere estimarla se debe optar por otro método que requiera menos información. Un método que puede ser útil es el de interpolación.

³La norma L_p para un vector v está dada por $\|v\|_p = (\sum_1^n |x_i|^p)^{1/p}$. Para $p \rightarrow \infty$ se obtiene la norma infinito $L_\infty = \max_i |v_i|$.

2.8.4. Paso de Proyección a interpolación

Como es de conocimiento, la representación óptima de un elemento en un subespacio para norma L_2 está dada por

$$h(\lambda, x) \approx \mathcal{P}_n[h](\lambda, x) = \sum_{j=1}^n c_j(\lambda) e_j(x)$$

con

$$c_j(\lambda) = \langle h(\lambda, \cdot), e_j(\cdot) \rangle,$$

que es la representación estándar por proyección. Usualmente se utilizan polinomios como base, aunque se pueden realizar elecciones más interesantes, que deriven en mejores representaciones. Se debe notar que para realizar una aproximación por proyección se necesita tener conocimiento de la función a representar en una cantidad grande de puntos para poder calcular computacionalmente de una forma precisa los coeficientes de proyección c_j .

En cambio, en la aproximación por interpolación se utiliza la expresión

$$h(\lambda, x) \approx \mathcal{I}_n[h](\lambda, x) = \sum_{j=1}^n C_j(\lambda) e_j(x),$$

donde los coeficientes $C_j(\lambda)$ son soluciones del problema de interpolación.

2.9. Método de Interpolación Empírica (EIM)

Al realizar una representación de una función con una interpolación, es de conocimiento que la elección de los nodos incide sustancialmente en el poder predictivo del interpolante. A partir de esto surge la necesidad de elegir los nodos de una forma eficiente. Como solución a esto se utiliza el algoritmo EIM [4] [7] [15] [21], que es un método que indentifica un conjunto de nodos cuasi-óptimo para realizar una interpolación a través de una base definida. Este algoritmo puede ser utilizado para realizar interpolaciones en dominios tanto unidimensionales como multidimensionales.

Características:

- Es un proceso con enfoque Greedy, dual a la construcción de una base reducida con un algoritmo Greedy.
- Para el elemento i -ésimo de la base, se construye un interpolante con los $i - 1$ nodos EIM ya encontrados y los primeros $i - 1$ elementos de la base. El i -ésimo nodo EIM estará dado por el punto del dominio donde el interpolante construido peor representa al i -ésimo elemento de la base.
- En cada iteración se minimiza la constante de Lebesgue, lo que permite que la finalidad del algoritmo sea satisfactoria.
- No necesita un dominio continuo para elegir los nodos interpolantes $\{T_i\}_{i=1}^n$, sino que elige dentro de los valores conocidos $\{t_i\}_{i=1}^L$.
- Por construcción del algoritmo, si se utiliza una base con n elementos, se obtendrán n nodos EIM $\{T_i\}_{i=1}^n$.

- Si se necesita mayor o menor cantidad de nodos, esto se consigue agregando o quitando nodos. Esto es una ventaja ante otros conjuntos de nodos, que aunque sean óptimos, si se requiere una cantidad mayor, se debe calcular un conjunto completamente nuevo.
- Al obtener los nodos más relevantes se puede realizar una compresión, es decir, utilizar una cantidad menor del total de datos temporales disponible, sin perder una cantidad significativa de precisión.

Algorithm 3 Algoritmo EIM

```

1: Input:  $\{e_i(x)\}_{i=1}^n$ ,  $\{x_i\}_{i=1}^L$ ,
2:  $X_1 = \operatorname{argmax}_x \|e_1(x)\|$ 
3:  $\text{nodos} = \{X_1\}$ 
4: for  $i = 2 \rightarrow n$  do
5:   Build  $\mathcal{I}_{i-1}[e_i](x)$ 
6:    $r = \mathcal{I}_{i-1}[e_i](x) - e_i(x)$  ( $r$  refiere al residual)
7:    $X_i = \operatorname{argmax}_x |r|$  ( $\operatorname{argmax}_x |r| = \|r\|_\infty$ )
8:    $\text{nodes} = \text{nodes} \cup \{X_i\}$ 
9: Output:  $\text{nodes EIM } \{X_i\}_{i=1}^n$  e interpolante  $\mathcal{I}_n(x)$ 

```

La aproximación EIM de una función $f(\lambda, x)$ está dada por la interpolación de la misma en los nodos EIM $\{X_i\}_{i=1}^n$, a partir de la base $\{e_i(x)\}_{i=1}^n$ utilizada para obtener los nodos:

$$f(\lambda, x) \approx \mathcal{I}_n[f](\lambda, x) = \sum_{j=1}^n B_j(x) f(X_j, \lambda)$$

con

$$B_j(x) = \sum_{i=1}^n e_i(x) (V^{-1})_{ij},$$

donde V_{ji} es la matriz de Vandermonde, dada por $V_{ji} = e_i(X_j)$.

2.9.1. Poder predictivo del algoritmo EIM

Como es de conocimiento, al realizar una interpolación, en la gran mayoría de los casos la representación obtenida es de menor precisión que una proyección. Además se sabe que la precisión de un interpolante con respecto a su contraparte de proyección está determinada por la constante de Lebesgue. A partir de la relación

$$\|h(\lambda, \cdot) - \mathcal{I}_n h(\lambda, \cdot)\|^2 \leq \Lambda_n \|f - \mathcal{P}_n h(\lambda, \cdot)\|^2,$$

se obtiene

$$\max_{\lambda} \|h(\lambda, \cdot) - \mathcal{I}_n h(\lambda, \cdot)\|^2 \leq \Lambda_n \max_{\lambda} \|h(\lambda, \cdot) - \mathcal{P}_n h(\lambda, \cdot)\|^2 = \Lambda_n \sigma_n.$$

La asignación de \max_{λ} en esta última ecuación conlleva cierta ambigüedad, ya que se puede referir al máximo con respecto a los parámetros del espacio de entrenamiento del algoritmo Greedy para la base reducida o a todo el intervalo de parámetros tratado. Esta elección depende del caso particular que se trate, pero el último resultado aplica a ambos casos. Lo mismo para la constante de Lebesgue.

Notar que la cota para el error obtenida muestra que la aproximación que se puede lograr es cuasi-óptima, ya que la constante de Lebesgue es minimizada por algoritmo EIM y σ_n decae rápidamente con n , como consecuencia de la elección Greedy de la base reducida.

2.10. Modelo predictivo a partir del interpolante empírico

A partir de los parámetros Greedy $\{\Lambda_i\}_{i=1}^n$ y sus funciones asociadas $\{h(\Lambda_i, t)\}_{i=1}^n$, el interpolante empírico para un parámetro λ dado queda definido como

$$h(\lambda, t) \approx \mathcal{I}_n[h](\lambda, t) = \sum_{j=1}^n B_j(t) h(\lambda, T_j),$$

con

$$B_j(t) = \sum_{i=1}^n h(\Lambda_i, t)(V^{-1})_{ij} \text{ y } V_{ji} = h(\Lambda_i, T_j).$$

En caso de conocer $\{h(\lambda, T_i)\}_{i=1}^n \forall \lambda$, $\mathcal{I}_n[h](\lambda, t)$ quedará determinado $\forall \lambda$ y por lo tanto se podrá construir una predicción de $h(\lambda, t) \forall \lambda, t$.

Como se tiene conocimiento de $\{h(\Lambda_j, t)\}_{j=1}^n$, en particular para cada nodo empírico T_i se dispone de los pares $\{(\Lambda_j, h(\Lambda_j, T_i))\}_{j=1}^n$, con los cuales se puede realizar regresiones para cada tiempo empírico T_i en el espacio de parámetros y obtener expresiones $\{h(\lambda, T_i)\}_{i=1}^n \forall \lambda$.

Si $h(\lambda, t) \in \mathbb{C}$, como sucede en el caso de ondas gravitacionales, a partir de la expresión

$$h(\lambda, t) = A(\lambda, t)e^{i\phi(\lambda, t)}$$

se puede realizar regresiones para fase y amplitud por separado para cada T_i . De esta forma serán necesarias $2 \times \dim(rb)$ regresiones (notar que la dimensión de la base reducida es igual a la cantidad de nodos empíricos).

Por lo tanto, para cada T_i , con los datos $\{A(\Lambda_j, T_i)\}_{j=1}^n$ y $\{\phi(\Lambda_j, T_i)\}_{j=1}^n$ en los nodos $\{\Lambda_j\}_{j=1}^n$ se pueden obtener regresiones $A_{fit}(\lambda, T_i)$ y $\phi_{fit}(\lambda, T_i)$, de forma que

$$h(\lambda, T_i) \approx A_{fit}(\lambda, T_i)e^{i\phi_{fit}(\lambda, T_i)}.$$

Reemplazando esta expresión en el interpolante empírico, se obtiene la expresión final del modelo predictivo $h_S(\lambda, t)$

$$h_S(\lambda, t) := \sum_{i=1}^n B_i(t) A_{fit}(\lambda, T_i) e^{i\phi_{fit}(\lambda, T_i)}, \quad \forall t, \lambda.$$

Este modelo fue introducido en el paper [19].

Una medida de la precisión del modelo está dada por $\|h(\lambda, \cdot) - h_S(\lambda, \cdot)\|^2$. Cuando se trata el problema de forma computacional, $h(\lambda, t)$ es discreto en el tiempo, por lo tanto es de utilidad la expresión discreta

$$\|h(\lambda, \cdot) - h_S(\lambda, \cdot)\|^2 := \Delta t \sum_{i=1}^L |h(\lambda, t_i) - h_S(\lambda, t_i)|^2,$$

donde $\Delta t := (t_{max} - t_{min})/(L - 1)$. y L es la cantidad de muestras en el tiempo.

Se puede mostrar que una cota superior al error está dada por

$$\Delta t \sum_{i=1}^L |h(\lambda, t_i) - h_S(\lambda, t_i)|^2 \leq \Lambda_n \sigma_n + \Lambda_n \Delta t \sum_{i=1}^n |h(\lambda, T_i) - h_S(\lambda, T_i)|^2,$$

donde $h(\lambda, T_i) - h_S(\lambda, T_i)$ proviene de la suboptimalidad de las regresiones realizadas.

Capítulo 3

Metodología y Resultados

3.1. Datos utilizados

Se utilizan ondas gravitacionales generadas a través del modelo Surrogate *SpECq1-10NoSpin*, construido a través de simulaciones de sistemas binarios de agujeros negros sin spin, obtenidas a través de Spectral Einstein Code (SpEC) [1] [27]. El modelo se obtiene a través del paquete en Python *gwsurrogate* [2].

El modelo provee ondas gravitacionales dentro del intervalo temporal $[-2750, 100]M$, donde se ha ubicado $t = 0$ en el instante de colisión de los agujeros negros. El espacio de parámetros que se trata es $q \in [1, 10]$.

Se proveen las polarizaciones h_+ y h_\times , por lo cual si se obtiene la expresión compleja $h = h_+ + ih_\times$, el espacio de funciones en cuestión se puede describir como

$$h(t, q) : t \times q \rightarrow \mathbb{C}$$

con $t \in [-2750, 100]M$ y $q \in [1, 10]$.

Este trabajo se centra en describir el modo dominante $l = m = 2$. En la figura 3.1 se muestran las polarizaciones h_+ y h_\times para $q = 2$ provistas por el modelo utilizado.

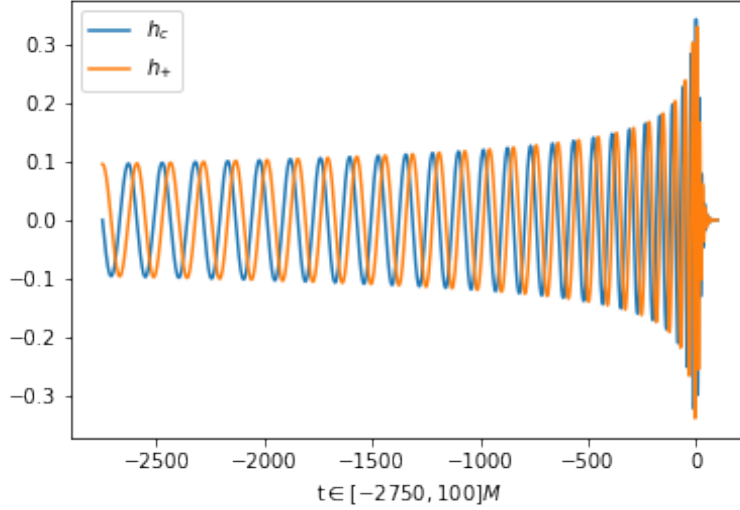


Figura 3.1: Polarizaciones h_+ y h_\times para $q = 2$, obtenidas con el modelo SpECq1-10NoSpin.

Para obtener el modelo predictivo se utilizó un conjunto de entrenamiento de 50 ondas gravitacionales, con parámetros distribuidos de forma equiespaciada en el espacio de parámetros. Para tener un conjunto de validación se generó un conjunto de 500 ondas, también con parámetros $q \in [1, 10]$ equiespaciados.

El modelo *SpECq1-10NoSpin* tiene un error relativo $\|\delta h(q)\|^2/\|h(q)\|^2$ máximo y mínimo aproximados de 1×10^{-4} y 9×10^{-6} respectivamente y establece un límite al cual se puede mejorar un modelo sustituto construido a partir de este. Cabe mencionar que este modelo Surrogate para la fase de regresión utiliza Least Squares para fase y amplitud, con polinomios de grado 5 en la variable $\nu = q/(1+q)^2$.

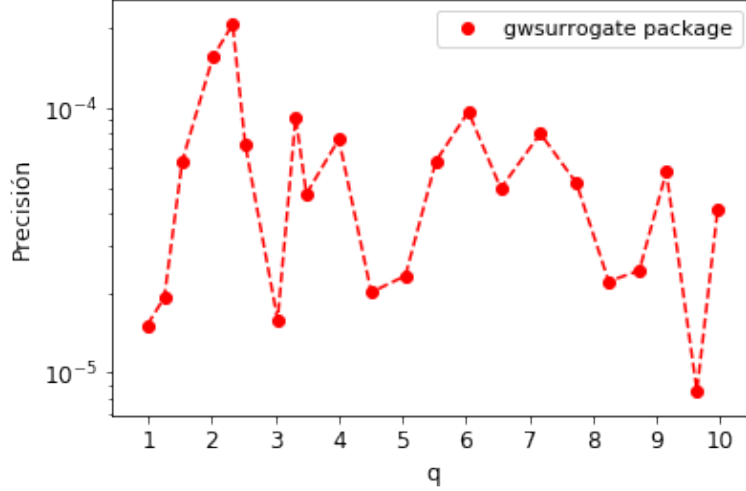


Figura 3.2: Error relativo $\|\delta h(q)\|^2 / \|h(q)\|^2$ de *SpECq1-10NoSpin* en función del espacio de parámetros.

Si se construye un modelo predictivo a partir de *SpECq1-10NoSpin* y se supera numéricamente la precisión de este modelo, no resulta de utilidad la magnitud en que se lo ha superado, ya que en caso de superar la precisión se está identificando el comportamiento del error numérico del modelo subyacente, es decir, se estaría realizando overfitting. Por lo tanto es indistinto si el modelo obtenido lo supera en precisión en uno o más ordenes de magnitud. Obtener overfitting en este caso no significa algo negativo, sino que se debe tener en cuenta hasta qué orden se otorgan resultados precisos.

Lo que resulta de interés es igualar o mejorar la precisión del modelo para todo parámetro. En caso de que ocurra esto, los modelos subyacente y sustituto (construido a partir de una cantidad finita de muestras del anterior) son idénticos en cuanto a precisión.

3.2. Algoritmo Greedy

Para obtener una base reducida $\{e_i\}_{i=1}^n$ y los parámetros más relevantes en el espacio de parámetros $\{\Lambda_i\}_{i=1}^n$ se construyó un algoritmo Greedy con el lenguaje Python, utilizando las librerías numpy y gwsurrogate, donde la última provee el modelo a partir del cual se obtiene el espacio de entrenamiento $\mathcal{T} := \{h_{\lambda_1}, \dots, h_{\lambda_N}\}$.

El espacio de entrenamiento utilizado se constituyó por 50 elementos $h(\lambda, t) = h_\lambda$ equiespaciados en el espacio de parámetros $\lambda := q, q \in [1, 10]$.

En cuanto al desarrollo del código del algoritmo Greedy, a continuación se mencionan los procedimientos utilizados:

- En la iteración n del algoritmo se realiza una proyección de cada $h_{\lambda_i} \in \mathcal{T}$ en la base $\{e_i\}_{i=1}^n$ construida hasta esa iteración, dada por $\mathcal{P}_n h_{\lambda_j} = \sum_{i=1}^n \langle e_i | h_{\lambda_j} \rangle e_i$. Para obtener los coeficientes de proyección $\langle e_i | h_{\lambda_j} \rangle$ se utilizó una cuadratura de Riemann para computar las integrales. Para aminorar el tiempo de convergencia del algoritmo se debe tener en cuenta que en la iteración $n + 1$ se realizarán nuevas proyecciones de los $h_{\lambda_i} \in \mathcal{T}$ sobre la nueva base $\{e_i\}_{i=1}^n \cup \{e_{n+1}\}$, por lo cual para evitar calcular de forma repetida los mismos

coeficientes de proyección se deben almacenar las proyecciones calculadas en la iteración anterior, de forma que en la nueva iteración solo sea necesario calcular un solo coeficiente por cada proyección. De esta forma la complejidad es lineal en la cantidad de bases, en lugar de combinatorial en el problema original.

- Para evitar acumulación de error numérico que impida la ortonormalidad de la base reducida resultante se utilizó el algoritmo de Gram-Schmidt Iterado, que logra sortear dicho inconveniente. En los casos de interés, a medida que se agrega elementos a la base reducida en el algoritmo Greedy, el método de Gram-Schmidt estándar muestra una creciente falla en la ortogonalidad de los elementos de la base resultante, lo cual invita a descartarlo.

Se exigió que el algoritmo Greedy otorgue una base reducida que represente a todas las funciones del espacio de entrenamiento con una precisión $\|h(q, \cdot) - \mathcal{P}_n h(q, \cdot)\|^2 < \epsilon = 10^{-16}$, donde n es la dimensión de la base reducida obtenida. Se utiliza dicho valor de ϵ con el fin de obtener una base reducida que permita representar a los elementos del espacio de entrenamiento de la mejor forma posible dentro de lo permitido por la aritmética de punto flotante de la computadora. Se utilizó como semilla del algoritmo a la onda gravitacional correspondiente a $\lambda = q = 1$. Los parámetros Greedy Λ_i obtenidos se muestran en la figura 3.3. La base reducida resultó de dimensión $n = 26$.

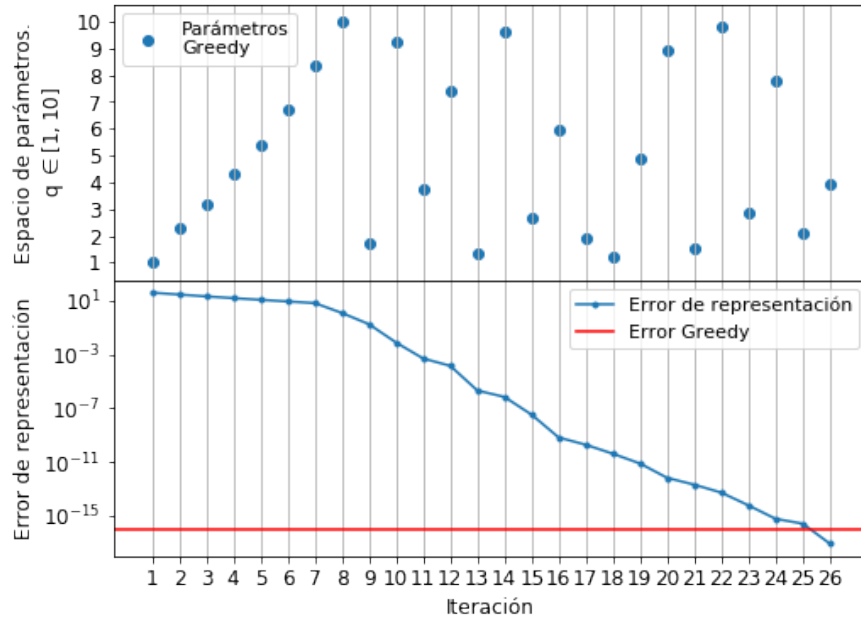


Figura 3.3: Arriba: Parámetros Greedy seleccionados por el algoritmo Greedy. En cada iteración se busca el mejor parámetro para enriquecer la base reducida, por lo cual el orden denota jerarquía entre ellos. Un nodo elegido en un paso dado tiene mayor jerarquía que los subsiguientes. Abajo: Convergencia del error de representación del espacio de entrenamiento para cada iteración del algoritmo Greedy, que finaliza cuando se sobrepasa la precisión requerida. El error es estrictamente decreciente a medida que se agregan nuevos elementos a la base, y eventualmente de convergencia exponencial, como previsto por la teoría.

Se compararon los resultados obtenidos con el algoritmo Greedy del paquete *rompy* [3]. Los parámetros Greedy obtenidos con *rompy* coinciden con los obtenidos, excepto que *rompy* agrega incorrectamente un elemento extra a la base reducida de forma innecesaria, cosa que ha sido resuelta con el código creado. Ocurre esto en *rompy* ya que el error de representación utilizado para comparar con ϵ al momento de iniciar un loop, corresponde al error de representación de la base obtenida en una iteración previa a la correspondiente. También se visualiza que *rompy* en diversas situaciones no logra obtener una base reducida con precisión ϵ , sino que finaliza antes de llegar a esta precisión, lo cual no es lo esperado para este trabajo.

3.3. Algoritmo EIM

Para realizar predicciones de ondas gravitacionales se realizan interpolaciones en el tiempo, y para que estas sean eficientes, se deben elegir nodos interpolantes apropiados. El algoritmo EIM a través de un enfoque Greedy provee un conjunto de nodos cuasi-óptimo. Este tiene muy buena convergencia ya que en cada iteración se enfoca en minimizar la constante de Lebesgue, que da una cota superior a la bondad de la precisión del interpolante (dada por el cociente del error del interpolante y la mejor aproximación en la base utilizada, dada por la proyección ortogonal). Al minimizar la constante de Lebesgue se impide que la imprecisión del interpolante sea grande.

Se obtuvieron los valores más relevantes del intervalo temporal $[-2750, 100]M$ con $\Delta t = 0,1M$ utilizando la base reducida obtenida con el algoritmo Greedy. Por lo tanto los nodos interpolantes serán elegidos del conjunto $\{-2750, -2750.1, \dots, 100\}$. Por construcción del algoritmo EIM el número de nodos está dado por la cantidad de elementos de la base. Los nodos obtenidos se visualizan en la siguiente figura.

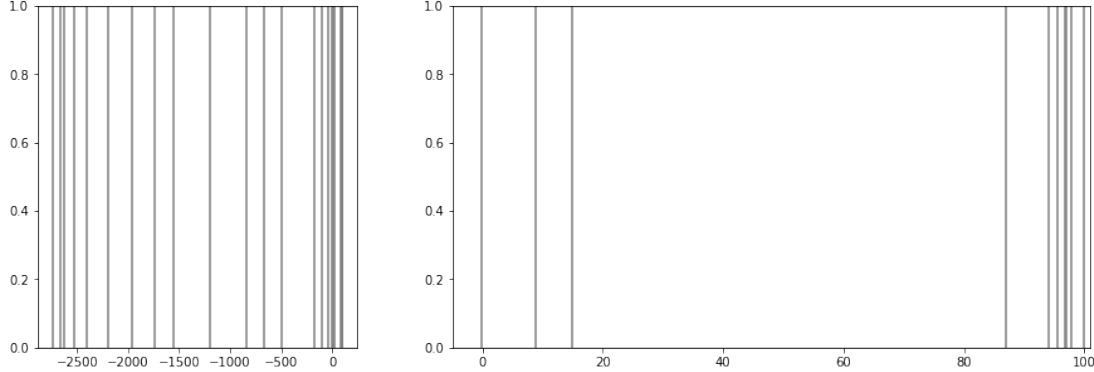


Figura 3.4: Distribución de los nodos EIM en el intervalo $[-2750, 100]M$ con $\Delta t = 0,1M$, utilizando la base reducida encontrada a través del algoritmo Greedy. La mayor acumulación de nodos se concentra en $t \approx 100M$, que físicamente se corresponde al *ringdown*, posterior a la colisión de los agujeros negros.

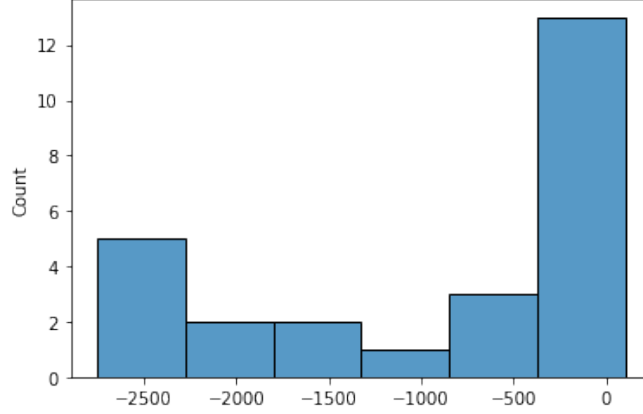


Figura 3.5: Histograma de la distribución de nodos EIM en el intervalo temporal $[-2750, 100]M$.

Se visualiza mayor acumulación de nodos en valores cercanos a $t \approx 100M$. También se observa una distribución similar a la de Chebyshev (en el aspecto que se encuentra mayor acumulación de nodos en los extremos), que es un buen indicio ya que estos son una buena elección al momento de realizar interpolaciones y de hecho son la elección óptima para interpolaciones polinomiales.

3.4. Fase de regresión

Luego de obtener los parámetros Greedy $\{\Lambda_i\}_{i=1}^n$ y los nodos EIM $\{T_i\}_{i=1}^n$, para cada T_j se realizan regresiones con los valores de las ondas contribuyentes a la base reducida $\{h(\Lambda_i, T_j)\}_{i=1}^n$ para estimar $h(\lambda, T_j) \forall \lambda$.

Se puede tener una noción del aporte del error de las regresiones a la precisión del modelo a través de una cota para su error, dada por

$$\|h(\lambda, \cdot) - h_S(\lambda, \cdot)\|^2 \approx \Delta t \sum_{i=1}^N |h(\lambda, t_i) - h_S(\lambda, t_i)|^2 \leq \Lambda_n \sigma_n + \Lambda_n \Delta t \sum_{i=1}^n |h(\lambda, T_i) - h_S(\lambda, T_i)|^2.$$

Se puede identificar contribuciones de dos fuentes. El primer término describe qué tan bien el interpolante empírico representa a $h(\lambda, t)$ a través de la elección de la base reducida y los nodos empíricos. A partir del decaimiento exponencial del error Greedy σ_n y del leve crecimiento de la constante de Lebesgue Λ_n , este término aporta muy poco a la cota. El segundo término está relacionado a la calidad de las regresiones realizadas. No se espera que la calidad de la regresión tenga una convergencia similar a la de σ_n , por lo cual resulta que el error de las regresiones es la fuente dominante de el error del modelo. Los resultados hasta la fecha corroboran esta afirmación. Como consecuencia de lo mencionado, se tienen motivos para realizar un estudio de la etapa de regresión del modelo para dilucidar nuevos métodos que ayuden a mejorar el poder predictivo del mismo.

Si se realiza una descomposición en fase y amplitud de los datos, donde

$$h(\Lambda_i, T_j) = A(\lambda_i, T_j) e^{i\phi(\Lambda_i, T_j)},$$

se puede visualizar que las funciones correspondientes a fase y amplitud son suaves y disponen de relativa simplicidad, por lo cual es menos costoso realizar regresiones de precisión a partir de

fase y amplitud que de la onda compleja. En un problema si se puede representar a los datos de una forma simple, es conveniente realizar regresiones bajo dada representación, ya que el modelado de los datos será un problema menos complejo. Por ejemplo, en Machine Learning es habitual utilizar Kernels con la finalidad de obtener representaciones más simples de los datos que faciliten el aprendizaje.

Si se descompone en fase y amplitud, se deben encontrar $2n$ funciones: $\{A_{T_i}(\lambda)\}_{i=1}^n$ y $\{\phi_{T_i}(\lambda)\}_{i=1}^n$. Luego la aproximación de las ondas complejas para cada T_i estará dada por

$$h(\lambda, T_i) \approx A_{T_i}(\lambda) e^{i\phi_{T_i}(\lambda)}.$$

Luego, para obtener la precisión de un modelos sustituto, se puede utilizar el error relativo de la onda para cada parámetro q , dado por

$$\frac{\|h(q, \cdot) - h_S(q, \cdot)\|^2}{\|h(q, \cdot)\|^2},$$

con

$$\|h(q, \cdot)\|^2 = \int dt |h(q, t)|^2.$$

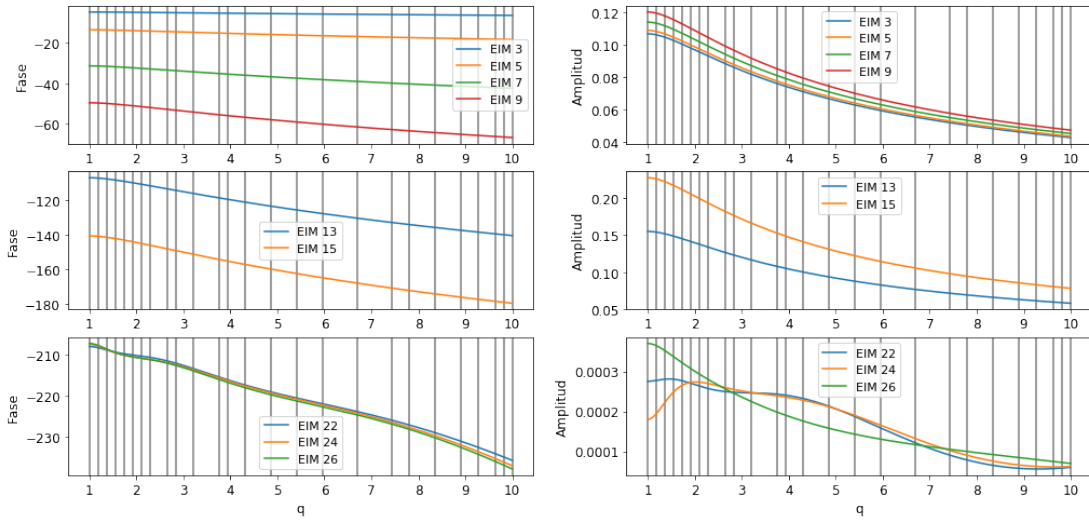


Figura 3.6: Conjunto representativo de fases y amplitudes en diferentes tiempos empíricos. Las líneas verticales indican la ubicación de cada uno de los parámetros Greedy Λ_i en el espacio de parámetros. Notar la concentración de nodos Greedy en $q \lesssim 2.5$ condice con el rango de mayor disimilitud entre las amplitudes de las ondas, donde mayor diferencia se visualiza entre las funciones graficadas para fase y amplitud a diferentes tiempos. Por esto es probable que el algoritmo Greedy ubique nodos en esa zona.

Ante la idea de encontrar una forma de predecir el comportamiento del comportamiento de fase y amplitud en los nodos empíricos se puede realizar un estudio a través de diferentes métodos de regresiones, tanto de de Machine Learning como de teoría de aproximación.

3.4.1. Regresiones con Machine Learning Automatizado

Las técnicas provenientes de Machine Learning (ML) han mostrado gran capacidad de aprendizaje y predicción de patrones complejos de datos a lo largo de gran cantidad de situaciones, por lo cual es natural que se guarde atención en ellas al momento de realizar las regresiones correspondientes al modelo sustituto.

Siempre al afrontar un problema a través de ML, existe una instancia en la que se decide el modelo a utilizar. Al momento de realizar la decisión, es común utilizar una cantidad limitada de modelos para luego elegir el mejor, o tomar decisiones bajo sesgos cognitivos y tender a preferir los que a priori se piense que otorgarán mejores resultados. Por ejemplo, enfoques como el de redes neuronales para problemas de ML puede parecer apropiado para gran cantidad de ocasiones, debido a los grandes resultados que se han obtenido con ellas en Data Science, pero esto no indica que para todo caso una red neuronal dada sea el mejor modelo a utilizar [23].

Por otro lado, a partir del principio de *No Free Lunch* para métodos de aprendizaje supervisado [29] [25] se sabe que ningún algoritmo de aprendizaje supervisado puede ser igualmente eficiente para aprender el comportamiento de todo conjunto de datos. Es decir que a priori no se puede saber cual será el más adecuado para realizar un modelado de un conjunto de datos determinado.

Ante estas situaciones se propone utilizar un conjunto amplio y diverso de modelos para luego elegir el mejor entre ellos, en lugar de utilizar los que a priori parezcan ser los más acertados. La búsqueda se puede realizar a forma de competición entre los algoritmos, en la cual se evalúa la capacidad predictiva de cada uno bajo una métrica dada para luego elegir el ganador. De esta forma se puede mitigar posibles problemas originados por sesgos cognitivos o por la dificultad de obtener un modelo eficiente para describir el comportamiento de los datos en cuestión.

También se debe tener en cuenta que puede requerir gran esfuerzo realizar el estudio de docenas de modelos por separado, donde a priori no se sabe cual será el de mayor optimalidad para un problema dado. Para poder realizar este estudio se realizó un acuerdo académico para utilizar la plataforma de AutoML DataRobot (DR) [20], capaz de automatizar y acelerar el proceso de ML, desde los datos crudos (es decir, sin ningún tratamiento previo) hasta el cómputo de predicciones de nuevos valores. DR se basa en la aplicación de buenas prácticas para los diferentes procedimientos de ML con el fin de encontrar modelos predictivos eficientes.

La plataforma DR permite la creación de un ranking de docenas de modelos a partir de diferentes métricas, relacionadas al de poder y velocidad de predicción. Diferentes procesos de ML pueden ser creados a través del mismo modelo, utilizando diferentes métodos de preprocesamiento de datos. DR se encarga de la búsqueda de hiperparámetros de los modelos y además se permite entrenar nuevos modelos con hiperparámetros especificados por el usuario.

A través de DR se puede tratar problemas de aprendizaje supervisado como no supervisado. Para este trabajo, en particular es de interés realizar regresiones. DR utiliza métodos de regresión clásicos, Procesos Gaussianos, Support Vector Machines, Gradient Boosted Trees, redes neuronales, algoritmos genéticos, entre otros. Se utilizan modelos open source como de propietarios tales como Eureqa¹.

DR permite el acceso al usuario a través de una plataforma web, donde se puede monitorear los modelos, realizar visualización de los datos y de los resultados con diversas métricas. También posee una API para Python, donde el usuario se puede abstraer de la página web y especificar

¹En el apéndice se lista todos los modelos usados por DataRobot.

los pasos a seguir por DR al realizar el entrenamiento, selección y evaluación de modelos. Los cómputos son realizados a través de instancias de Amazon Web Services (AWS), que es un servicio de cómputo, almacenamiento y base de datos en la nube provisto por Amazon.

Para cada regresión se utilizan más de 120 Blueprints distintos, definidos por DR como el proceso que recorren los datos, desde el preprocesamiento utilizado hasta el modelo de ML (por ejemplo dos modelos de ML idénticos, donde en uno se realiza una normalización de los datos y en otro no se consideran como diferentes Blueprint).

Para cada modelo sustituto, en el caso de trabajo son necesarias 52 regresiones, donde 26 son para la fase de las ondas, y las 26 restantes para amplitud. Cada una es entrenada con 26 datos y solo 26 datos, i.e. solo los elementos de la basa reducida, cada uno correspondiente a uno de los parámetros Greedy. En este sentido, uno de los desafíos de este trabajo es trabajar con un conjunto, en la etapa de regresión, minimalista.

Para elegir los mejores modelos se utilizaron dos métricas diferentes: 5-Fold Cross Validation² con los datos de entrenamiento y el error cuadrático medio (RMSE) con el conjunto de validación. En este caso, Cross Validation (CV) es de utilidad en caso que no se pueda generar un conjunto de validación para testear el poder predictivo del modelo en datos diferentes a los de entrenamiento. Por otro lado, el error sobre el conjunto de validación de 500 ondas es representativo del poder predictivo de los modelos en todo el espacio de parámetros, ya que este es lo suficientemente denso como para cubrirlo.

Las métricas utilizadas tienen la finalidad de predecir el poder predictivo de los modelos, por lo tanto, el mejor modelo sustituto para cada métrica estará construido con el mejor modelo para cada regresión, obtenido con dicha métrica. En caso de querer utilizar un único modelo de ML para todas las regresiones, los mejores modelos sustitutos será obtenidos a través de los modelos de ML que mejor puedan modelar todas las regresiones.

En el caso de K-Fold CV se utilizó $K = 5$, con la métrica RMSE. Cada una de las K muestras fue formada por datos tomados al azar. Se realiza esto para que cada uno de los K modelos creados para obtener el score sea entrenado con datos provenientes de todo el dominio. Si se tomase por ejemplo 5 muestras donde cada una contiene datos de $[i, i + 2]_{i=1, \dots, 5}$, los modelos resultantes pueden tener mala performance en intervalos donde no se encuentran datos de entrenamiento y el score de CV puede no proveer una buena predicción de la capacidad de aprendizaje del modelo.

Se obtuvo un ranking de todos los modelos de DR para cada una de las regresiones según el score de 5-Fold CV. A continuación se muestra un histograma de los diferentes modelos que han liderado un ranking en al menos una regresión:

²K-Fold Cross Validation: métrica para cuantificar la capacidad de predicción de un modelo sobre datos con los cuales no ha sido entrenado. Los datos se dividen de forma aleatoria en k conjuntos de tamaño similar (en lo posible del mismo tamaño). Un modelos se entrena con $k-1$ conjuntos y con el restante se estima el poder predictivo del mismo. Se obtienen k modelos de forma que cada uno se valida con uno de los k subconjuntos. A través de una métrica dada se obtiene el poder predictivo de cada muestra y se toma la media aritmética de los resultados. El valor obtenido representa al poder predictivo del modelo. Al tomar muestras aleatorias de los datos este score tiene un bias, para cada sampleo, el valor obtenido será diferente.

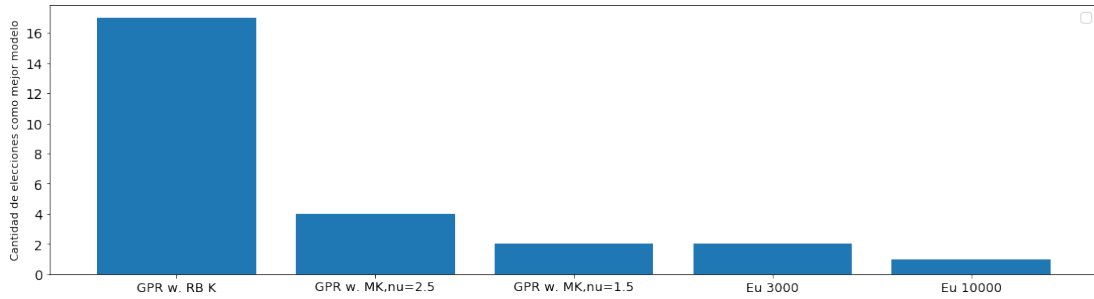


Figura 3.7: Modelos ganadores de DataRobot para alguna de las regresiones correspondientes a la fase de las ondas.

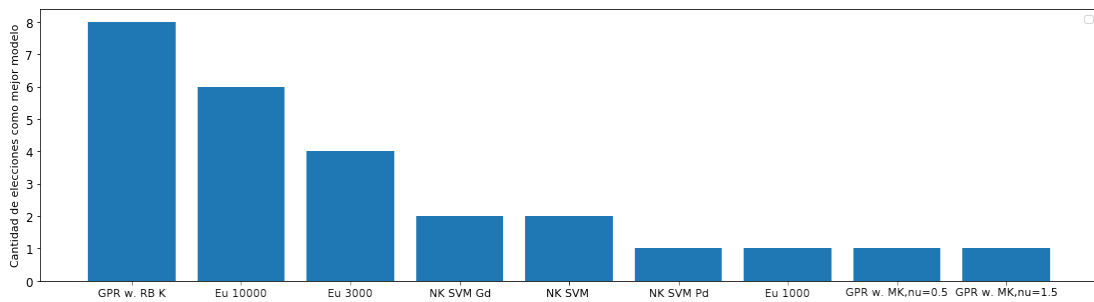


Figura 3.8: Modelos ganadores de DataRobot para alguna de las regresiones correspondientes a la amplitud de las ondas.

Notar que siempre, de más de 120 posibilidades, se han elegido los mismos 5 modelos para fase y los mismos 7 modelos para amplitud. Esto se debe a la similitud de los datos para cada caso. Si para conjuntos de datos con comportamientos muy similares se obtuviera performances muy diferentes para los mismos modelos resultaría cuanto menos un hecho curioso.

Al observar las regresiones ganadoras de fase y amplitud en conjunto, se puede notar que el modelo que más veces ha sido el ganador es la regresión de Procesos Gaussianos (GP) con Kernel de Bases Radiales (RBK) [28], un modelo que en general ha sido desechado históricamente por la comunidad de ondas gravitacionales sin hacer un estudio exhaustivo como el presente.

Los modelos que han sido ganadores al menos una vez, han mostrado superioridad sobre los otros empleados por DR no solo por haber sido los ganadores en al menos una ocasión, sino que también en gran mayoría de los casos han sido los que han liderado los ranking de modelos.

Para las regresiones de amplitud, los modelos ganadores en las regresiones para la fase han sido dominantes. También se debe notar que para regresiones de amplitud han tomado mayor relevancia las regresiones Support Vector Machine (SVM), que han sido ganadoras en algunas ocasiones. En concreto esto sucede en nodos EIM del 20 al 24, debido a que los datos de estas regresiones son los que más difieren al resto, esto se ve claramente en la figura 3.6.

A partir del RMSE con el conjunto de validación, las mejores regresiones obtenidas se pueden ver en la imagen 3.9.

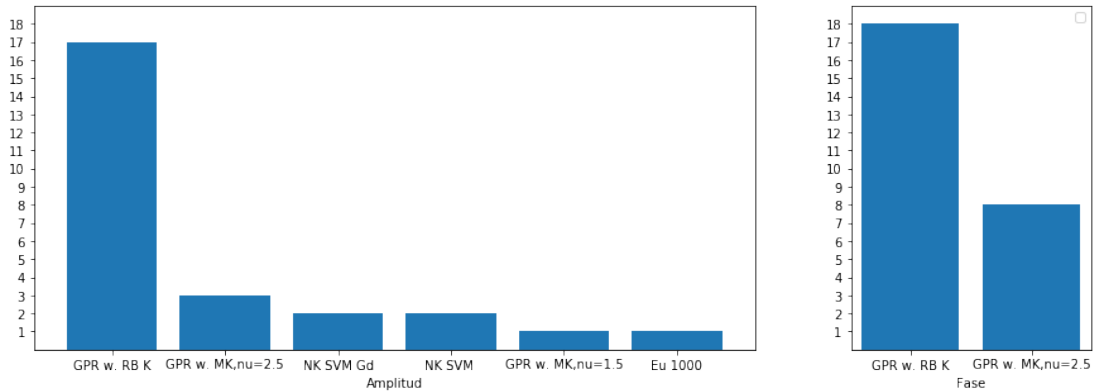


Figura 3.9: Modelos ganadores de DataRobot a partir de RMSE con el conjunto de validación.

Excepto las regresiones de amplitud en nodos EIM del 21 al 16, siempre han sido ganadores los modelos GP con RBK y Matern Kernel (MK) con $\nu = 2, 5$. En amplitud y nodos del 21 al 26 resultan como ganadores modelos SVM, similar a lo ocurrido con 5-Fold CV, debido a que los datos en los nodos EIM de tiempos grandes ($t \approx 100M$) son los que más difieren al resto, ver imagen 3.6.

Si se tiene que elegir un único modelo de ML a través del conjunto de validación, se elegirá GPR con RBK, ya que ha sido el mejor modelo por un gran margen. Por otro lado, el mejor modelo predictivo se construye eligiendo la mejor regresión para cada nodo EIM, donde la métrica para hacer esto es RMSE sobre el conjunto de validación.

Luego de obtener los mejores modelos para cada regresión, se construyó un modelo sustituto por cada modelo ganador y otro utilizando en cada regresión el mejor a partir de las métricas 5-Fold CV y RMSE para el conjunto de validación. A través del error relativo para una onda compleja, se validó el poder predictivo de cada modelo a lo largo de todo el espacio de parámetros con una muestra de 500 ondas equiespaciadas, generadas con el mismo paquete que se creó el espacio de entrenamiento.

Se comparó el poder predictivo de cada modelo con la precisión máxima del modelo subyacente, donde en caso de obtener un error relativo menor que este $\forall q \in [1, 10]$, se habrá creado un modelo sustituto sin pérdida de poder predictivo.

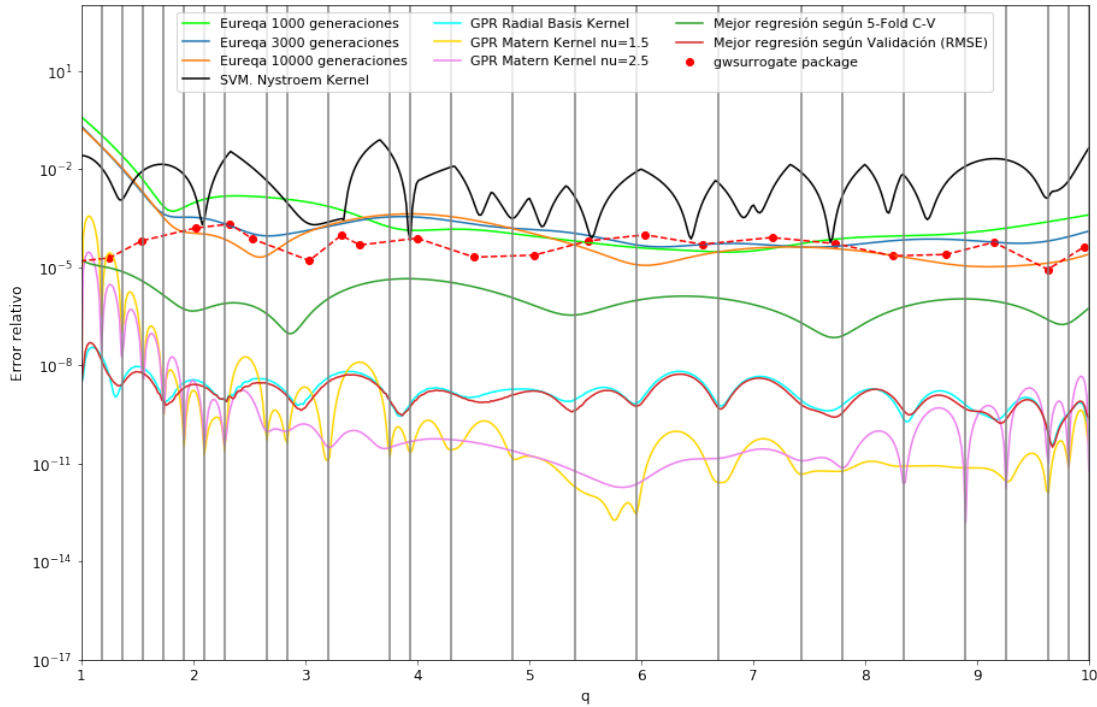


Figura 3.10: Error relativo de las predicciones de los modelos sustitutos creados a través de DataRobot.

Los modelos obtenidos con Eureqa (regresión simbólica usando programación genética), donde cada uno utiliza diferente número de generaciones, en ciertos intervalos del espacio de parámetros logran superar la precisión del modelo fidedigno, aunque lo óptimo es que esto ocurra para todo q . En particular para $q \lesssim 2$ la performance que se obtiene a través de ellos no es la mejor, ya que el error relativo crece cuando $q \rightarrow 1$ y es de hasta de más de 3 ordenes de magnitud mayor que el del modelo fidedigno. La performance obtenida a través de SVM tampoco ha sido satisfactoria, ya que no logra superar al modelo subyacente en ninguna ocasión.

Los modelos obtenidos con GP y MK con $\nu = 1,5$ y $\nu = 2,5$ han sobrepasado ampliamente los valores obtenidos del paquete en gran parte del espacio de parámetros, aunque su precisión decrece a medida que $q \rightarrow 1$ y para $q \approx 1$ no logran replicar exactamente al modelo fidedigno.

Los modelos obtenidos que logran superar la precisión del modelo fidedigno son los obtenidos a través del ganador para cada regresión y el construido a través de GP con RBK.

Cabe mencionar que GPR con bases radiales fue el modelo de ML que más veces resultó ganador en la etapa de regresión con el score CV, lo cual muestra que con esta métrica se pudo predecir de buena forma un buen modelo para realizar todas las regresiones.

El modelo con los ganadores para cada regresión con el conjunto de validación es el mejor en cuanto a error medio, ya que los modelos fueron elegidos a través de RMSE. Comparando con el construido solamente con GP y RBK, se puede ver que este no es el modelo donde el error tiene menor norma infinito, esto es consecuencia a que no fue calculado para ser el mejor bajo esta métrica. Si se quiere encontrar el modelo con menor error máximo, se debe elegir las regresiones no con RMSE, sino a través de la norma infinito.

Como se ha discutido, si se quiere obtener el mejor modelo sustituto posible, se debe elegir el

mejor modelo para cada regresión. Al hacerlo con 5-Fold CV no se obtiene mejor resultado que otros modelos. Esto se debe a que esta métrica no predice exactamente el poder predictivo de un modelo. Se puede buscar una solución a esto utilizando una mejor métrica para poder realizar una mejor elección de modelo. Posibles soluciones pueden ser utilizar otro K, como puede ser Leave-One-Out-Cross-Validation (LOOCV) o realizar la elección de un modelo luego de obtener una cantidad mayor de scores de CV.

Si se quiere elegir el mejor modelo a través de 5-Fold CV, se obtendrá a GPR con RBK. Pero el segundo mejor se obtiene utilizando Eureqa con 10000 generaciones, que es uno de los mejores modelos, pero se puede ver a través del error con el conjunto de validación que no es el segundo mejor. Esta es otra muestra de la incerteza que tiene 5-Fold CV para elegir los mejores modelos. También se debe mencionar que en caso de elegir el mejor modelo sustituto con esta métrica se habría elegido al conformado por las mejores regresiones para dicha métrica y no al de GP con RBK, que al utilizar el conjunto de validación se puede ver que este último otorga mejores resultados.

Al construir un modelo sustituto, se puede elegir las regresiones tanto en la etapa de regresión como después de haber construido el modelo sustituto, ya que mejor regresión en etapa de regresión es equivalente a obtener un mejor modelo sustituto, utilizando métricas que permitan dilucidar el poder predictivo obtenido con las regresiones utilizadas.

En caso de no tener un conjunto de validación con el que se pueda predecir de buena forma el poder predictivo de los modelos, no siempre se tiene una métrica para saber cual regresión es la mejor. Esto queda reflejado en la incerteza de 5-Fold CV. Uno de los motivos que puede ocasionar no contar con un buen conjunto de validación es el costo computacional que puede implicar generarlo.

En resumen se ha podido ver qué regresiones son de utilidad para obtener un modelo sustituto preciso, en el sentido que ha superado la precisión del modelo subyacente utilizado y también se ha podido construir modelos precisos a través de 5-Fold CV y tener una noción de su incerteza.

3.4.2. Regresiones con modelos clásicos

Por otro lado se utilizaron modelos clásicos de regresión para obtener el modelo Surrogate. Se utilizaron los métodos de interpolación polinomial, Splines Cúbicos y LS con polinomios.

Se sabe que con interpolación y LS el aumentar la cantidad de datos de entrenamiento no implica una mejor regresión (fenómeno Runge), por lo tanto puede no ser óptimo entrenar las regresiones con todos los datos. En cuanto a LS, se debe tener en cuenta que al utilizar un espacio V_m con $\dim(V_m) = m$ para aproximar una función con n datos conocidos, m mayor no implica mejor resultado, en particular cuando $n \approx m$. Como consecuencia de esto, se obtuvieron modelos sustitutos variando con diferentes criterios la cantidad de nodos Greedy para entrenar las regresiones. En el caso de LS también se utilizaron espacios de diferentes dimensiones. Estos análisis además permiten tener noción de la convergencia de cada uno a medida que se cambian las condiciones sobre las cuales se obtienen.

Interpolación

La elección de los nodos Greedy al construir la base reducida es de forma jerárquica, es decir que los primeros n nodos Greedy elegidos denotan los n más relevantes en el espacio de parámetros en cuestión (notar que esto no cuenta para el primer nodo Greedy, que se elige de forma aleatoria, pero luego el segundo nodo es la mejor elección para el primero, lo mismo el tercer nodo para los primeros dos, y así sucesivamente). Por lo tanto si se quiere realizar una buena interpolación con un polinomio de grado n , es buena sugerencia elegir los primeros $n + 1$ nodos Greedy. Para ver la dependencia de los modelos con la cantidad de datos que se entrenan

las regresiones, se obtienen modelos para diferentes elecciones de nodos y se obtiene la capacidad predictiva de cada uno a través del conjunto de validación.

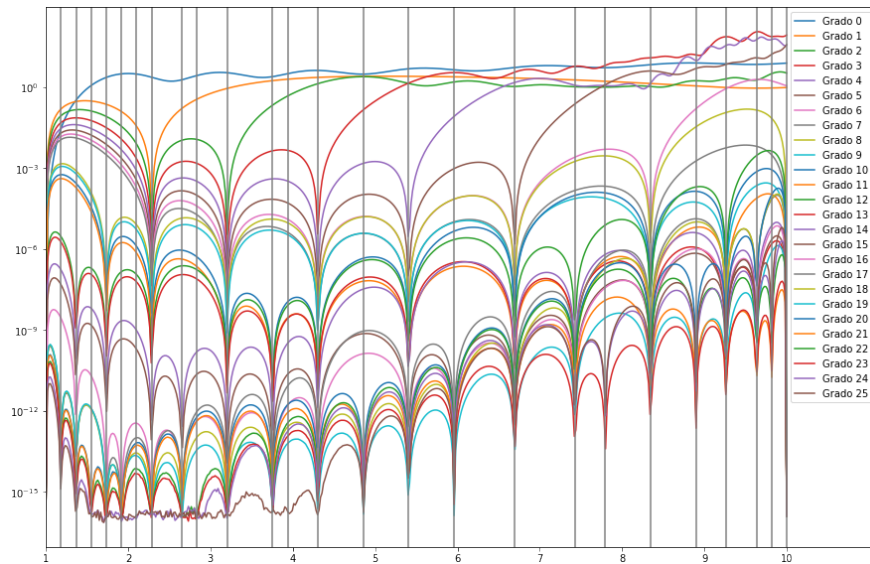


Figura 3.11: Errores relativos en el espacio de parámetros q para modelos sustitutos con polinomios interpolantes de diferente grado. Para interpolaciones de grado n se utilizaron los primeros $n + 1$ nodos Greedy. Cada línea vertical denota la ubicación de un parámetro Greedy.

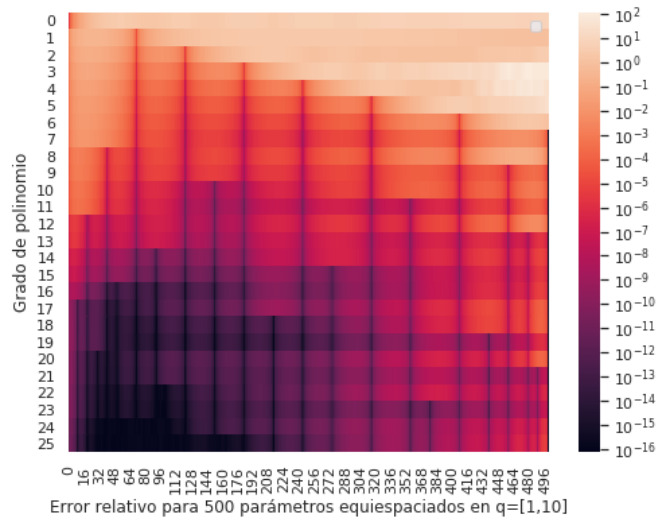


Figura 3.12: Heatmap de los errores relativos en función del espacio de parámetros y el grado de interpolante utilizado, donde para interpolaciones de grado n se utilizaron los primeros $n + 1$ nodos Greedy.

Para testear qué tan acertada es la decisión de tomar los nodos Greedy en cuanto a su jerarquía para realizar interpolaciones, se compararon los resultados con regresiones obtenidas a través de elecciones aleatorias de los nodos. La elección aleatoria también se utiliza de forma anidada (es decir, el caso de $m + 1$ nodos, corresponde al caso de m nodos con un nuevo nodos elegido de forma aleatoria).

También se realizaron regresiones con nodos diferentes a los Greedy para tener una noción de qué tan buenos son estos últimos para obtener regresiones. Se eligió un conjunto con la misma cantidad de datos que parámetros Greedy y se realizaron pruebas de convergencia eligiendo los nodos de forma anidada y aleatoria.

Se realizó un estudio de convergencia de los errores máximos y medios al aumentar la cantidad de nodos (y por lo tanto de grado de polinomio) con los diferentes criterios. Los mejores resultados fueron obtenidos a través de la elección de nodos Greedy en orden Greedy, ya que la convergencia que se visualiza para este caso es la mejor. La elección aleatoria de nodos Greedy no ha logrado superar la performance obtenida a través de la elección en orden Greedy. Esta es una evidencia de que la jerarquía de los nodos Greedy es útil al momento de elegir nodos para interpolar. Por último, como era de esperar, la peor convergencia se ha obtenido al realizar interpolaciones con nodos diferentes a los Greedy seleccionados de forma aleatoria, ya que es de conocimiento que para realizar interpolaciones precisas es necesaria una distribución de nodos que lo permita, en general esto pasa con distribuciones de nodos en las cuales la densidad de nodos aumenta hacia los límites del intervalo. Este comportamiento ocurre con la distribución de los nodos Greedy.

En cuanto al error máximo, el modelo de mayor precisión se obtuvo para una interpolación de los primeros 22 nodos Greedy. Con respecto al error medio, el mejor modelo se construyó a partir de los primeros 24 nodos Greedy. Ambos logran replicar al modelo subyacente en todo el espacio de parámetros.

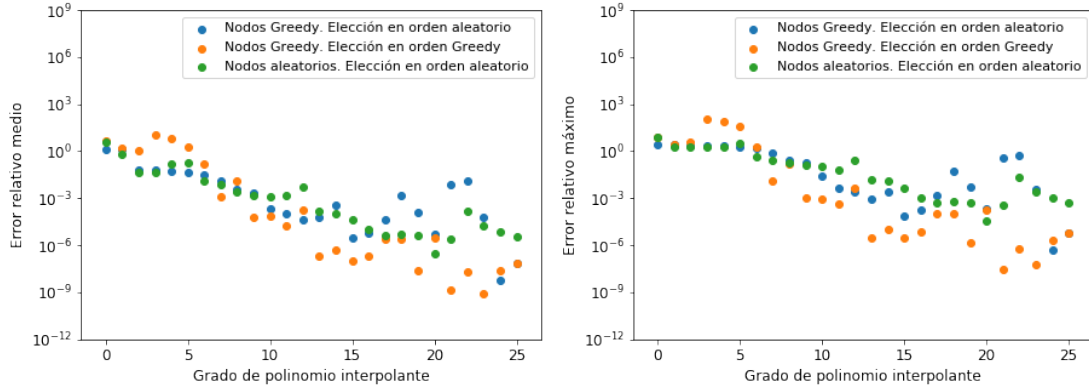


Figura 3.13: Convergencia de los errores relativos máximos y medios del modelo sustituto al agregar datos con diferentes criterios.

Cuadrados mínimos

Al momento de afrontar un problema de cuadrados mínimos (LS) hay dos métricas que son de importancia. Suponiendo que $A_{true}(q)$ es una función a reconstruir y $A_{fit}(q)$ como a su contraparte obtenida a través de cuadrados mínimos, una métrica es el error cuadrático medio (MSE)

$$\|A_{true}(q) - A_{fit}(q)\|_n^2 = \frac{1}{n} \sum_{i=1}^n |A_{true}(q_i) - A_{fit}(q_i)|^2,$$

y la otra métrica, que es de característica global, cuantifica el error en todo el volumen en el que se realiza la regresión

$$\|A_{true}(q) - A_{fit}(q)\|^2 = \int_q |A_{true}(q) - A_{fit}(q)|^2 dq.$$

Se afrontó la cuestión de encontrar el mejor modelo posible con LS a través de dos enfoques diferentes: uno donde se elige la mejor regresión en la etapa de regresión del Surrogate, comparándolas a través de MSE sobre los datos de entrenamiento, para luego construir un modelo sustituto, y otro donde se compara los modelos sustitutos obtenidos a través de un único modelo de LS, con un grado de polinomio fijo para todas las regresiones. En los dos casos se obtiene poder predictivo de cada modelo sustituto con el conjunto de validación, sobre las predicciones del mismo.

En el primer caso, para cada conjunto de datos de fase y amplitud en los parámetros Greedy $\{\Lambda_i\}_{i=1}^n$, se utilizó el conjunto de espacios $\{V_j\}_{j=1}^n$ con $V_j = \{x^i\}_{i=0}^{j-1}$ para resolver el problema de LS. Se eligió para cada regresión de fase y amplitud a la que menor error cuadrático medio tuvo sobre los datos de entrenamiento y luego se construyó el modelo Surrogate. Se valida con los datos de entrenamiento al momento de realizar las regresiones para tener una noción del resultado que otorgan, ya que en ocasiones no se dispone de un conjunto de datos diferente al obtenido a través de la base reducida, debido al costo computacional que implicaría generarlos.

Siempre se obtuvo como mejores polinomios a los de grado n al trabajar con $n + 1$ datos. Estos son los mejores resultados ya que para n datos en el problema de LS con polinomios de grado $n - 1$ es el caso de interpolación, donde se obtiene $MSE = 0$. El resultado obtenido en este apartado es el mismo que el obtenido al hacer interpolaciones con todos los nodos, visto en

la sección anterior de interpolación. Este resultado no implica que la regresión utilizada sea la mejor posible, ya que esta métrica solo depende de una cantidad acotada de datos en el intervalo entrenado, y lo que se quiere es que la regresión obtenida sea la mejor globalmente. Para que $MSE = 0$ sea un valor que indique la mejor regresión posible, tiene que ser comparable con la medida global, es decir $\|A_{true}(q) - A_{fit}\|_n^2 \approx \|A_{true}(q) - A_{fit}\|^2$. Notar que para $n \approx m$ se puede tener $MSE \approx 0$ con un error global muy grande, que correspondería a un overfitting de los datos.

Si se quiere elegir una regresión en la etapa de regresión del modelo, al medir la precisión de las regresiones con métricas discretizadas se debe prestar atención que la métrica utilizada sea comparable a el error global de la regresión, que es lo que importa para el modelo sustituto, ya que la regresión debe ser eficiente en todo el espacio de parámetros λ . En casos donde no se puede validar el modelo sustituto con un conjunto de datos que no forma parte de los datos de entrenamiento y se deba utilizar MSE o alguna otra métrica sobre los datos de entrenamiento, se debe tener en cuenta este resultado. Como mejor MSE no implica un mejor modelo, no es lo óptimo utilizarlo para dilucidar cual regresión es más apropiada.

Lo que se quiere es un mejor modelo sustituto, y por esto en el segundo caso no se procedió eligiendo la mejor regresión en la etapa de regresión, sino a través el error del modelo sustituto con el conjunto de validación, ya que no se tiene una buena métrica para realizar la elección de la mejor regresión al momento de la construcción del modelo. Se realizaron regresiones con un grado de polinomio fijo para cada modelo. En este caso se comparará la precisión de los modelos sustitutos resultantes y no la precisión de cada regresión. Como también incide en la bondad de las regresiones la cantidad de datos utilizados, se obtuvo diferentes modelos donde en cada uno se varia los datos de entrenamiento con diferentes criterios: con los primeros n nodos Greedy, con elección aleatoria de n nodos Greedy y con datos aleatorios, los mismos utilizados para el análisis con interpolaciones.

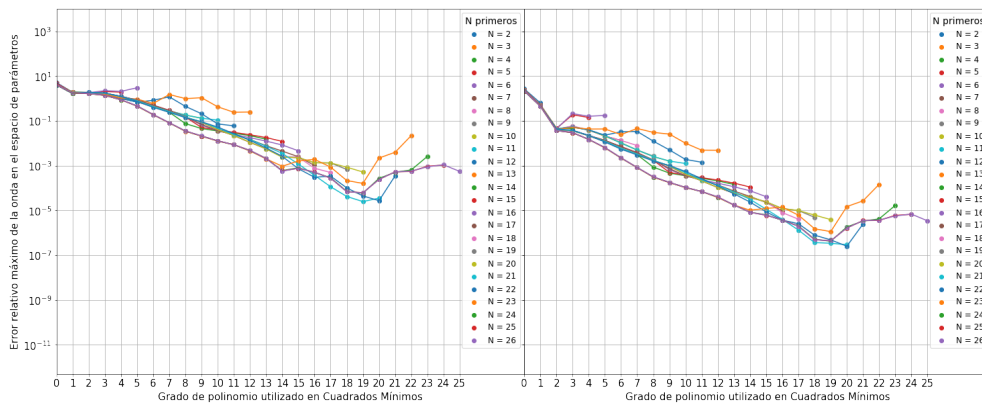


Figura 3.14: Errores máximos (izquierda) y errores medios (derecha) de modelos obtenidos con LS con nodos aleatorios, elegidos en orden aleatorio.

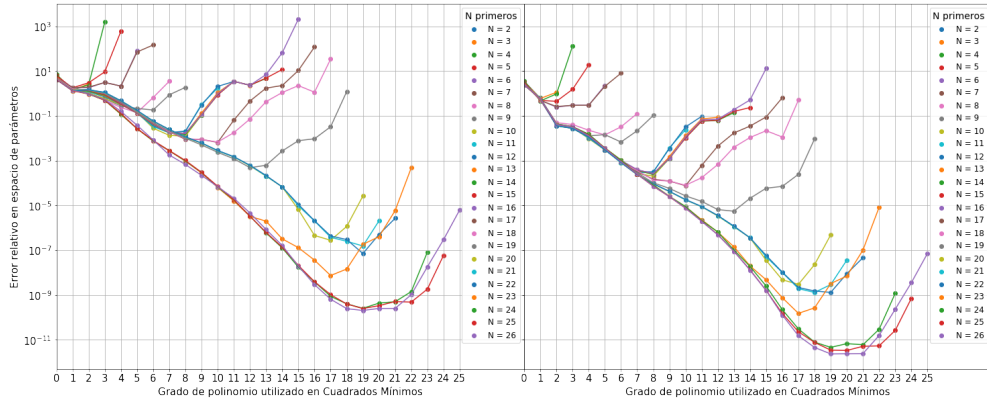


Figura 3.15: Errores máximos (izquierda) y errores medios (derecha) de los modelos sustitutos obtenidos con LS y nodos Greedy, elegidos en orden aleatorio.

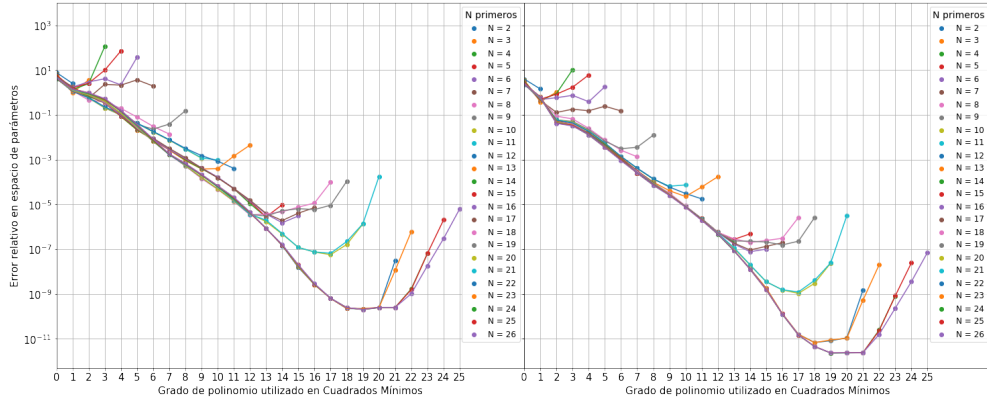


Figura 3.16: Errores máximos (izquierda) y errores medios (derecha) de modelos sustitutos obtenidos con LS y nodos Greedy, elegidos en orden Greedy.

Los nodos Greedy, elegidos inicialmente para la construcción de la base reducida, permiten que la capacidad de aproximación aumente varios órdenes de magnitud con respecto a si se tuviese nodos distribuidos de forma aleatoria. Además, si se realiza una elección aleatoria de los nodos Greedy, el error comienza a crecer de forma anticipada con respecto a una elección debida a la jerarquía de los nodos. Por lo tanto esta última posibilidad es la más indicada.

Se observa para los tres casos que el error aumenta para $n \approx m$, esto es esperable ya que es de conocimiento que para LS esta condición generalmente implica resultados no óptimos. Suele ser preferible que el grado de polinomio utilizado sea menor que la cantidad de nodos, que tan menor depende del caso particular [16].

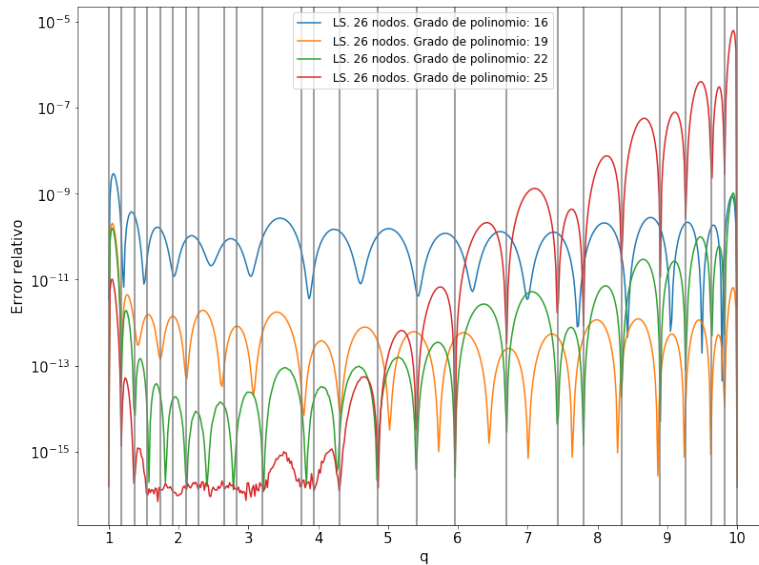


Figura 3.17: Errores en el espacio de parámetros utilizando LS y todos los parámetros Greedy. El error máximo en el espacio de parámetros no decrece a mayor grado de polinomio para todo caso.

En cuanto al error máximo, los mejores resultados fueron obtenidos para elección de nodos en orden Greedy con números de nodos mayores o iguales a 22 y para elección de nodos Greedy en forma aleatoria con 24 o más nodos, siempre utilizando polinomios de grado 19. En todos estos casos se ha obtenido errores relativos máximos de aproximadamente 2×10^{-10} .

Si se compara los resultados obtenidos al elegir el mejor modelo en la etapa de regresión y al mejor modelo elegido por el error del modelo sustituto, se puede ver que los mejores resultados se obtienen para el último caso, probando que la métrica MSE en la etapa de regresión no es óptima para obtener el mejor modelo.

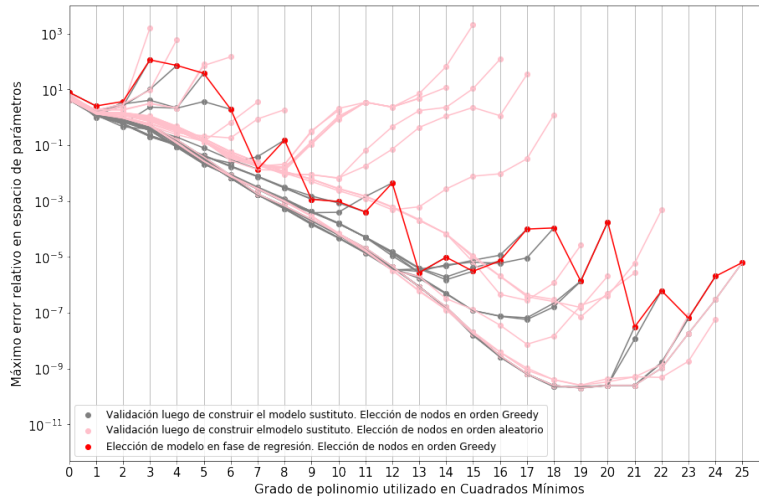


Figura 3.18: Comparación de errores obtenidos al elegir el modelo de LS en la etapa de regresión y a través del error del Surrogate.

Splines

Para realizar un estudio más amplio se utilizó regresiones locales, en particular Splines Cúbicos, con el método *not-a-knot* para determinar de forma unívoca los polinomios utilizados para construir las regresiones [17].

La convergencia obtenida no decae con la misma intensidad que para los modelos utilizados anteriormente. Esto se debe que Splines es un método donde se utiliza solamente polinomios cúbicos, que pueden realizar underfitting para comportamientos de datos complejos. Como se ha visto en las secciones de LS e interpolación, eventualmente polinomios de grado mayor permiten mejores representaciones. El beneficio que se obtiene al construir modelos con Splines es evitar la presencia del fenómeno Runge y el bajo costo de evaluación de la regresión, ya que se trabaja con polinomios cúbicos, y la cantidad de operaciones que se debe realizar para una evaluación es muy chica si se utiliza el algoritmo de Horner (6 operaciones exactamente).

Los mejores resultados se observan para la mayor cantidad de datos de entrenamiento. Splines al ser un método de regresión local, no genera problemas al aumentar la cantidad de nodos para entrenar la aproximación. Los mejores modelos superan la precisión del paquete *gwsurrogate*, por lo tanto con polinomios cúbicos para la fase de regresión se puede crear un modelo sustituto sin pérdida de precisión con respecto al del paquete. Notar que si el modelo subyacente tuviese mayor precisión puede que esto no ocurra.

Como se podía suponer, la mejor convergencia fue obtenida a través de elegir los n primeros nodos Greedy para interpolar, y los peores resultados se obtuvieron al realizar regresiones con nodos aleatorios.

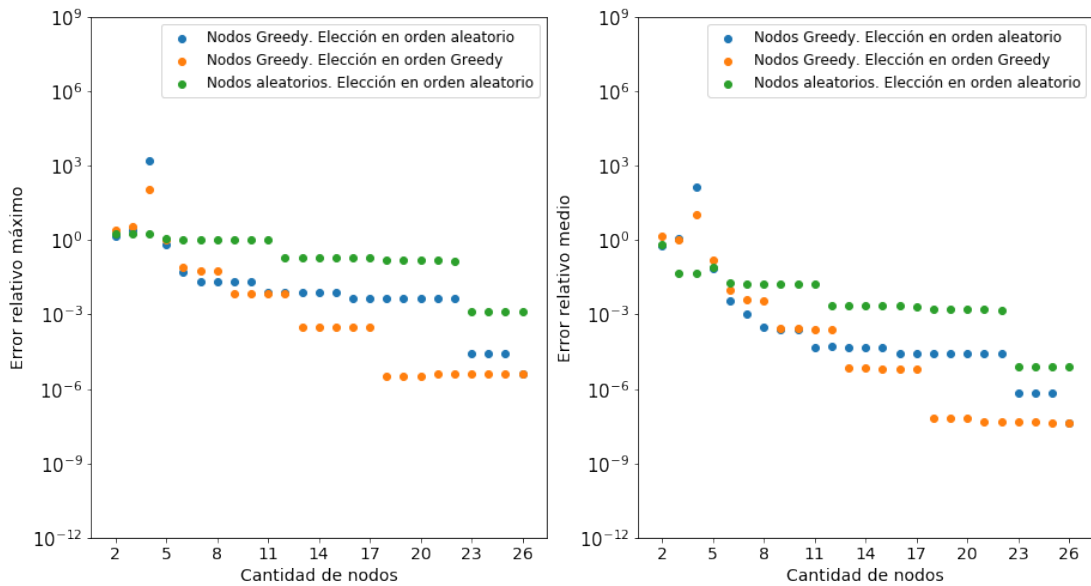


Figura 3.19: Convergencia utilizando Splines Cúbicos con diferentes elecciones de nodos.

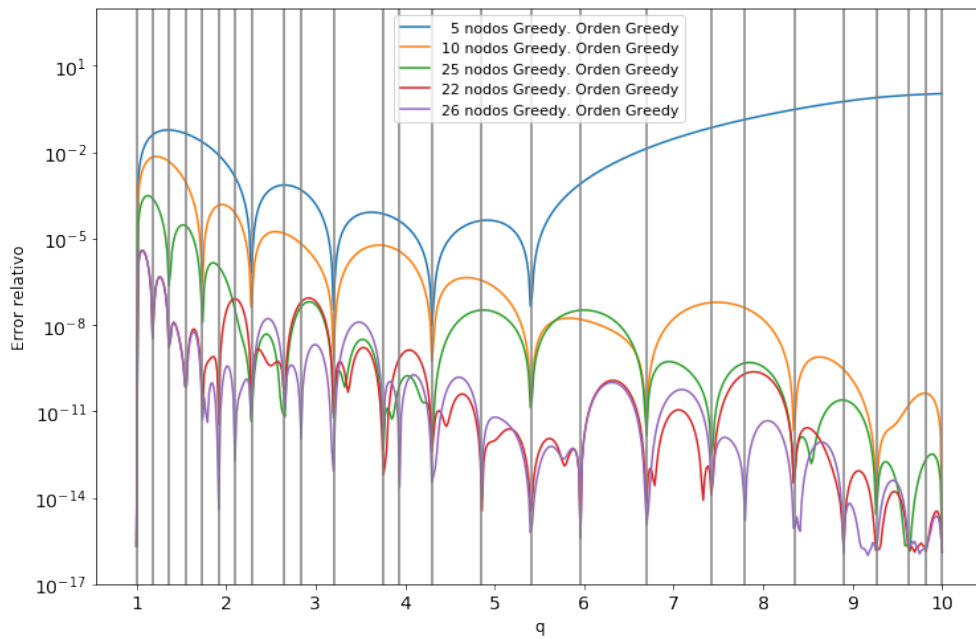


Figura 3.20: Error en el espacio de parámetros de modelos Surrogate obtenidos con Splines, utilizando diferentes cantidades de nodos.

Vecinos Cercanos

Las predicciones a través de vecinos cercanos (NN, por su denominación en inglés, Nearest Neighbors) son métodos de Machine Learning donde la predicción de un valor para un dato dado

se basa en las características de los datos más cercanos. No existe una única forma de definir una distancia, dependiendo de la elección de esta generalmente se obtienen diferentes resultados y a partir de esta libertad ocasionalmente se podrá sacar algún provecho.

Para utilizar este método de forma eficiente debe haber cierta correspondencia en el comportamiento de un dato con un conjunto de datos "vecinos". Para definir los vecinos se utiliza alguna métrica de conveniencia para cuantificar las distancias entre los datos.

El número óptimo de vecinos cercanos depende exclusivamente del comportamiento de los datos en cuestión. Por ejemplo si se define un conjunto de datos donde dentro de una esfera de radio r tienen un comportamiento descrito por una función dada, y suponiendo que es una constante por simplicidad, y fuera de ella el comportamiento de los datos es totalmente aleatorio. Si se quiere modelar el comportamiento de los datos dentro de la esfera, en caso de elegir los vecinos más cercanos dentro de un radio r para realizar una predicción, lo esperable es que el r óptimo sea menor o igual que el diámetro de la esfera, es decir, $2r$. Como r está definido por la distribución de los datos en el espacio, el número de vecinos cercanos óptimos depende directamente del comportamiento de los datos.

Realizar modelos con vecinos cercanos permite modelar localmente los comportamientos de los datos, cosa que no ocurre por ejemplo al momento de realizar regresiones globales, que son métodos predictivos donde se obtiene una única regresión para modelar el comportamiento en todo el dominio en cuestión.

Una de las motivaciones para utilizar regresiones a partir de NN es tratar de evitar los resultados obtenidos en los mejores modelos de interpolación, donde el error aumenta varios órdenes de magnitud para q grande, y en el caso de Splines también crece considerablemente para q chicos.

Se construyeron modelos sustitutos utilizando los n vecinos más cercanos al parámetro correspondiente de la onda a estimar, es decir que dependiendo el parámetro que se elija para estimar la onda correspondiente, la regresión realizada se obtendrá a través de diferentes conjuntos de datos. Es necesario definir una distancia para seleccionar los n primeros vecinos cercanos de cada dato, la más natural es la distancia euclídea³ y por ello esta se ha utilizado.

Explícitamente, para obtener una expresión de una solución $h(\lambda', t)$ con un modelo sustituto de n vecinos, se procede de la siguiente forma:

- Se buscan los n valores del espacio de parámetros correspondientes a datos de entrenamiento más cercanos a λ' (denótense como $\{\Lambda_i\}_{i=1}^n$).
- Para cada tiempo empírico T_j se obtienen expresiones $h_{fit}(\lambda, T_j)$ con los pares $\{(\Lambda_i, h(\Lambda_i, T_j))\}_{i=1}^n$.
- Para cada tiempo empírico T_j se evalúa $h_{fit}(\lambda, T_j)$ para obtener $h_{fit}(\lambda', T_j)$.
- Se construye la predicción $h_S(\lambda', t) = \sum_j B_j(t)h_{fit}(\lambda', T_j)$.

La cantidad de vecinos a utilizar es un hiperparámetro del problema, que debe ser definido antes de la etapa de entrenamiento de las regresiones. Como a priori no se tiene conocimiento del número de vecinos cercanos que otorgará el mejor modelo, se estudió todos los casos posibles. Para cada regresión se utilizan 26 datos, así que realizar una búsqueda entre todos los hiperparámetros es una tarea poco costosa computacionalmente hablando.

Se utilizó los mismos métodos de regresión clásicos para obtener predicciones a través de los vecinos cercanos: LS, interpolación polinomial y Splines Cúbicos.

³ Sea V un espacio vectorial y $v, w \in V$, entonces $d(v, w)^2 = \sum_{i=1}^n (v_i - w_i)^2$

Vecinos Cercanos con cuadrados mínimos

En el caso de LS, para n vecinos cercanos se creó n modelos sustitutos a partir de regresiones con polinomios de grado m para cada uno, con $m = 0, \dots, n - 1$.

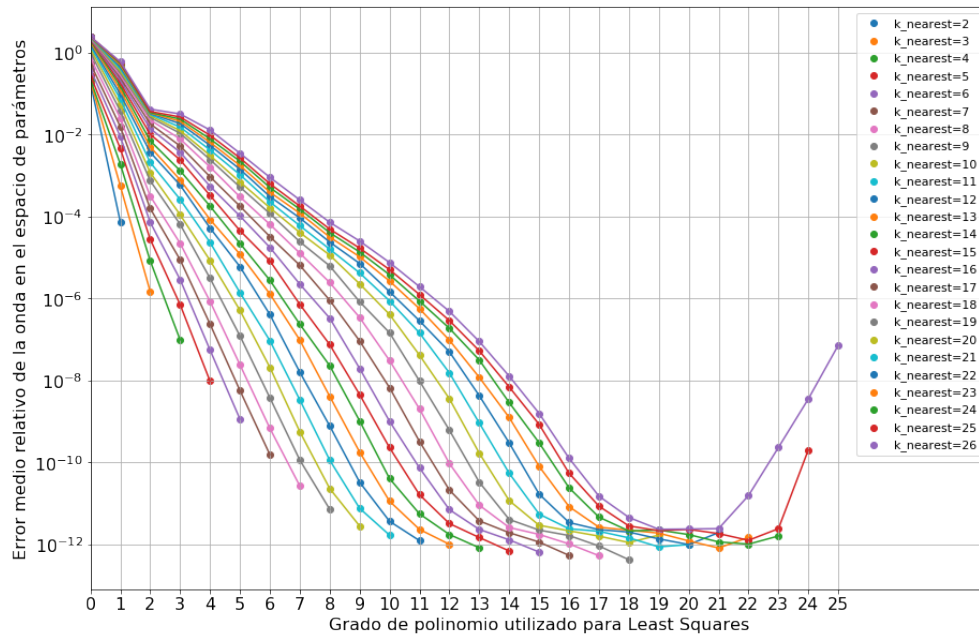


Figura 3.21: Error relativo medio de modelos Surrogate obtenidos utilizando NN y LS.

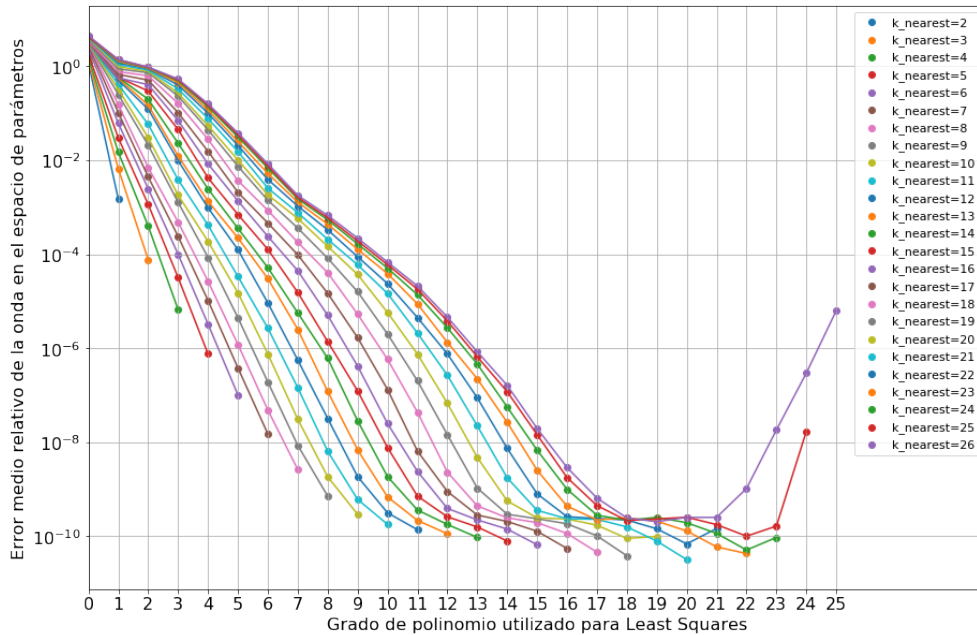


Figura 3.22: Error relativo máximo de modelos Surrogate obtenidos utilizando NN y LS.

Como se puede apreciar en los gráficos, en general tanto para el error máximo del Surrogate en todo el espacio de parámetros como para el error medio, para un polinomio de grado n dado se obtiene mejor resultado entrenándolo con los $n + 1$ valores más cercanos. Para números mayores de vecinos cercanos la precisión decrece, debido al fenómeno análogo al de Runge para interpolación. Otro patrón que se observa es que a medida se agrega datos de vecinos más lejanos para un grado de polinomio fijo, en la gran mayoría de casos los resultados empeoran.

Si se toma n vecinos cercanos, en general es mejor interpolar con polinomios de grado $n - 1$, excepto para n grande (cercano a 26, que es el valor máximo de vecinos cercanos que se puede disponer, debido a la cantidad de puntos Greedy con los que se trabaja).

Entre todos los modelos, el de menor error máximo se obtiene con 21 vecinos cercanos y con un polinomio de grado 20 (es decir que se construye con interpolaciones), otorgando un error máximo de $3,19 \times 10^{-11}$, superando a todos los modelos obtenidos con DR y regresiones globales.

Notar que el caso de 26 vecinos cercanos y grado 25 del polinomio utilizado es el caso en que se toma todos los datos disponibles para realizar cada regresión, es decir, se realiza una interpolación global.

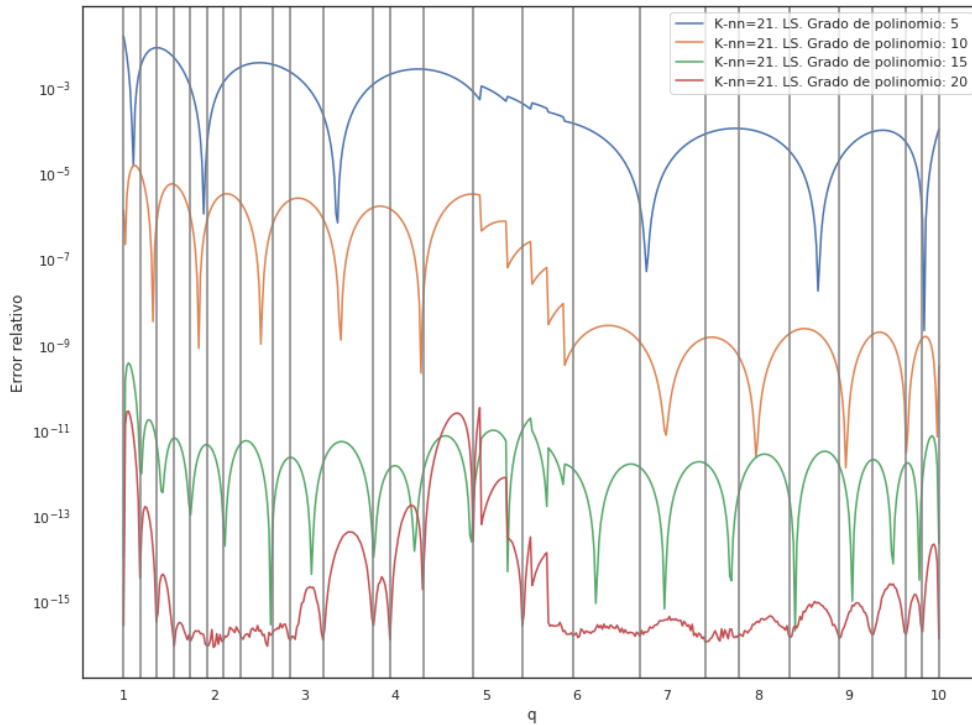


Figura 3.23: Error relativo en el espacio de parámetros para el modelo sustituto de mínimo error máximo. Las líneas verticales denotan la ubicación de los puntos Greedy.

Para el mejor modelo se puede ver que no se tiene crecimiento del error relativo para q grande, como pasa en el caso de interpolación para grados altos de polinomios y tampoco el error decrece notablemente para q chicos, como pasa con Splines. El error está mucho más controlado con este método. También se logró obtener modelos de errores máximo y medio menor con respecto a estos últimos.

Vecinos Cercanos con Interpolación

Se obtuvo modelos sustitutos con vecinos cercanos utilizando polinomios interpolantes como modelo de regresión. Como se mencionó anteriormente, este es el caso del problema de LS, donde el grado del polinomio usado es una unidad menor que la cantidad de datos utilizados para entrenar la regresión.

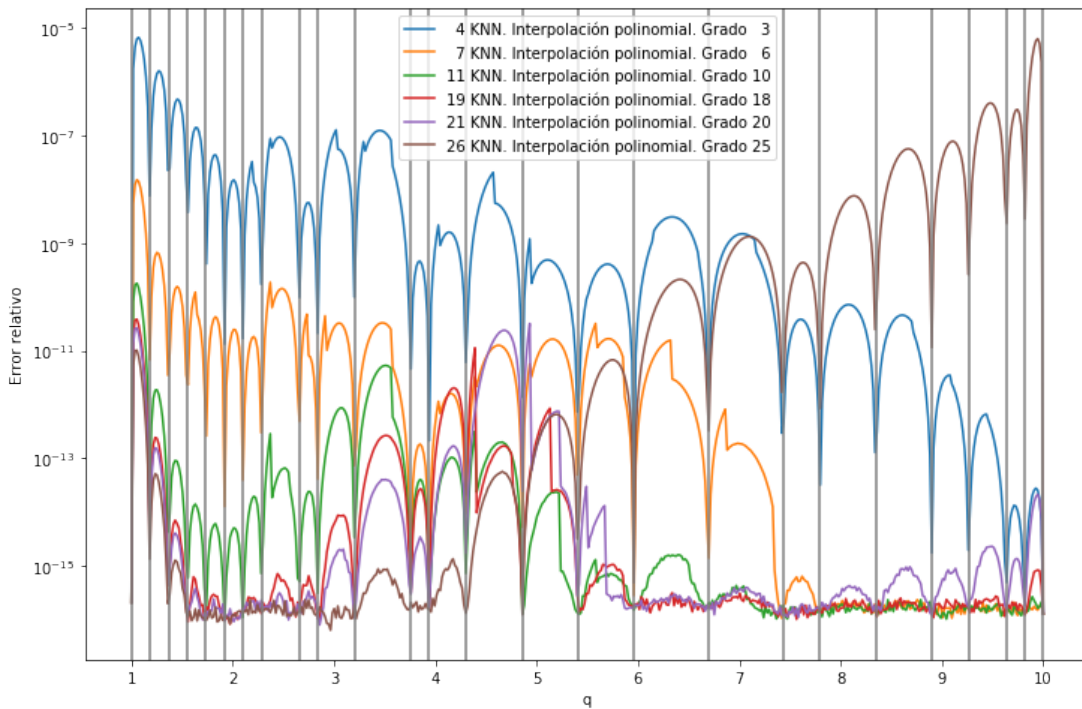


Figura 3.24: Error relativo de las predicciones en función del parámetro q de modelos obtenidos con NN e interpolación polinomial.

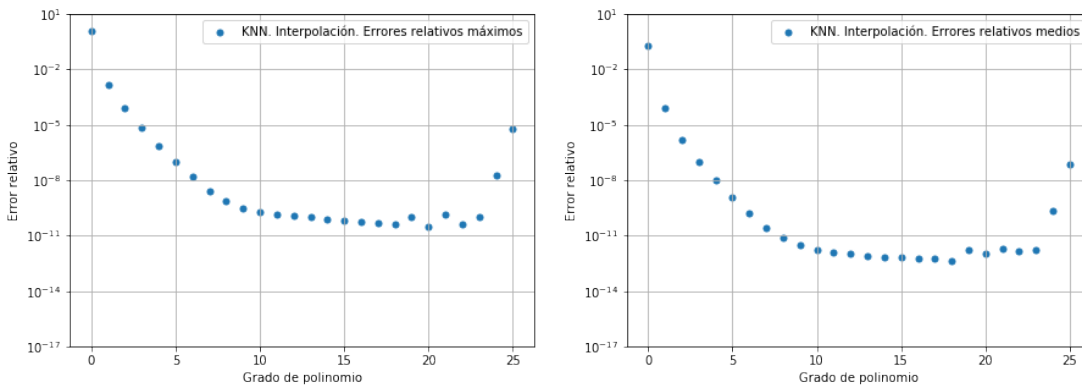


Figura 3.25: Errores relativos máximos y medios de modelos predictivos construidos con NN e interpolación polinomial.

En cuanto al comportamiento de errores a lo largo del espacio de parámetros, se puede ver que los resultados de polinomios más bajos (es decir con menor cantidad de vecinos cercanos) se asemejan a los obtenidos con Splines (polinomios interpolantes locales de grado 3).

El método de vecinos cercanos con interpolación es el que más rápido converge al aumentar el grado de polinomio, dentro de todos los enfoques vistos.

El costo de evaluación de un polinomio en general crece exponencialmente con la dimensión

del dominio. Si la dimensión d es 1, el costo de la evaluación es $\mathcal{O}(n^2)$ y a través del método de Horner el total de operaciones son $2n$. Para $d = n$, si se tiene hasta grado m en cada dimensión, el costo es $\mathcal{O}(m^{2n})$. Para realizar evaluaciones rápidas es necesario tener en cuenta estos conceptos. Aprovechando la buena convergencia de este método, si se quiere regresiones con grados de polinomio menor debido al menor costo de evaluación, se puede elegir un modelo con 4 vecinos cercanos, ya que supera la precisión del modelo fidedigno. Por ejemplo se puede ver que el error máximo del modelo de con 4 vecinos cercanos es similar al de interpolación con todos los nodos y utilizando el algoritmo de Horner el costo de evaluar una regresión del primero sería de $2*3 = 6$ operaciones y para el segundo de $2*25 = 50$, por lo cual es preferible el primero ante el segundo, debido a que la etapa de evaluación para el primer modelo sería menos costosa, lo cual es de gran importancia para modelos Surrogate, que se centran principalmente lograr evaluaciones rápidas y precisas. En este caso lo que se realiza es un tradeoff entre precisión y costo de evaluación.

Se obtuvo al modelo con menor error máximo utilizando 21 vecinos cercanos, donde el modelo predictivo ha tenido error de $3,19 \times 10^{-11}$. El mejor modelo en cuanto a error medio a lo largo del espacio de parámetros se obtuvo con 18 vecinos, con error de $4,25 \times 10^{-13}$.

Vecinos Cercanos con Splines

Se observó el comportamiento de los errores de los modelos obtenidos a través de Splines Cúbicos con K vecinos cercanos, para todo K . Al igual que en los apartados anteriores relacionados a Splines, se utilizó la condición *not-a-knot*.

El punto a favor que tiene Splines, como toda interpolación local, es la no existencia del fenómeno Runge. Por ejemplo, en un intervalo acotado si se interpola de forma local, al agregar nodos de forma aleatoria o equiespaciada, cubriendo todo el intervalo, el error decae a cero. En este caso el error no decae a cero porque se agregan nodos cada vez más lejanos, ampliando el intervalo sobre el cual se interpola.

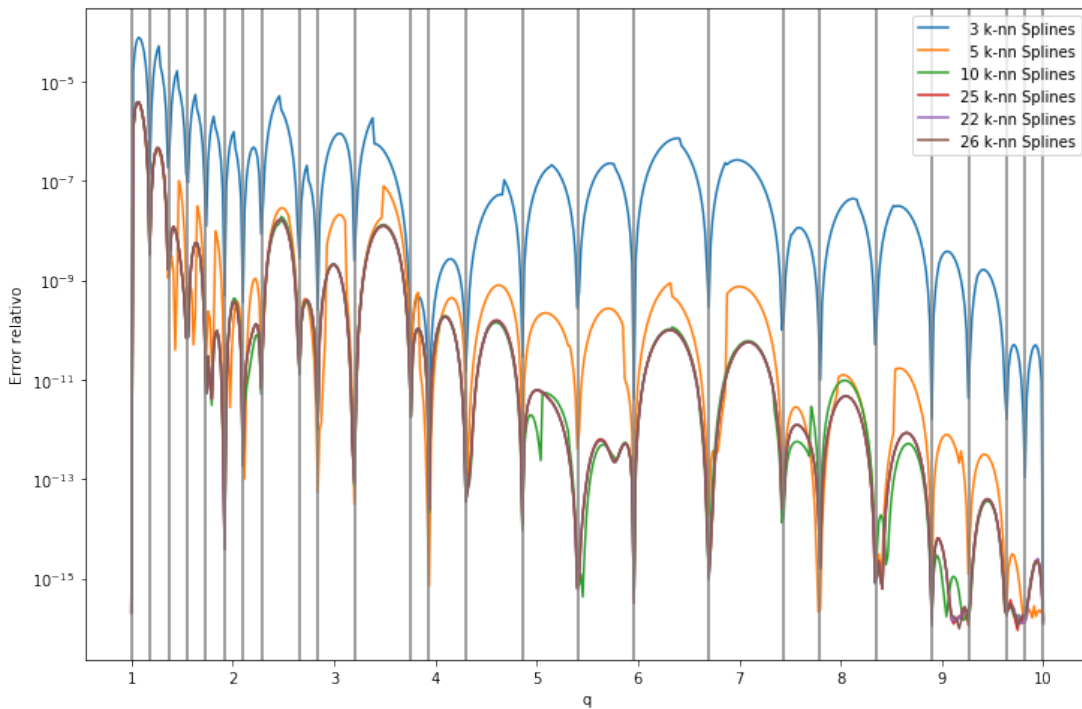


Figura 3.26: Error relativo en función del parámetro q de modelos obtenidos con Splines Cúbicos y diferentes números de vecinos cercanos.

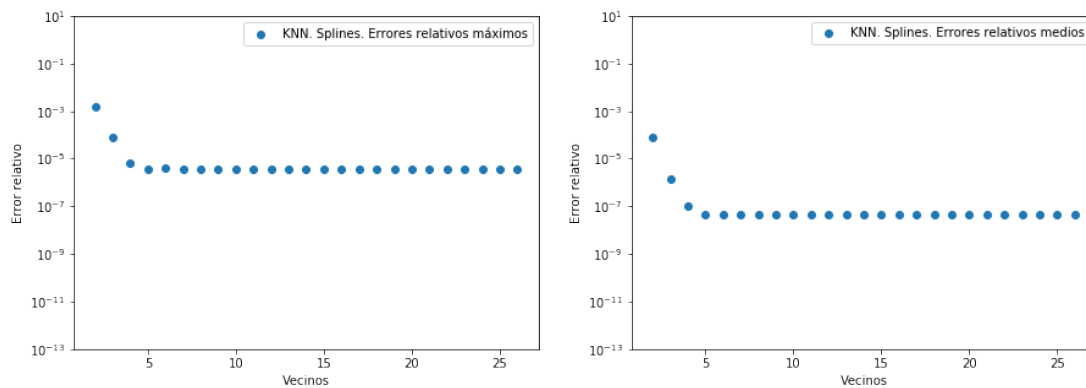


Figura 3.27: Errores relativos máximos y medios en el espacio de parámetros de modelos obtenidos con Splines Cúbicos y NN.

Agregar vecinos más lejanos no afecta de una forma considerable al valor de la regresión en el lugar donde se la evalúa. Esto se debe a que al ser una regresión donde los extremos de cada polinomio están fijos en sus extremos por los nodos y condicionados por la imposición de la continuidad de sus derivadas, la curva obtenida no dispone de una libertad considerable para cambiar su comportamiento. LS al no imponer una condición tan fuerte de fijar la curva en puntos dados, dispone de más versatilidad para poder modelar la curva final.

Este método no presenta resultados con ventajas sobre los mejores resultados obtenidos realizando interpolaciones globales con Splines, ya que las precisiones obtenidas no son mejores y el costo de obtener el modelo es mayor, ya que se realizan mayor cantidad de regresiones e incluso se deben computar los vecinos cercanos del parámetro correspondiente a la onda a estimar.

Capítulo 4

Conclusiones

Al construir un modelo sustituto hay dos finalidades: alto poder predictivo y velocidad de evaluación. En cuanto al poder predictivo, la instancia del modelo que más contribuye a la imprecisión de las predicciones es la etapa de regresión. En este trabajo se ha realizado un estudio comparativo de un amplio conjunto de regresiones con el fin de tener conocimiento de buenas prácticas para obtener modelos lo más precisos posibles. También se han reconocido los métodos de regresión más indicados.

Para emplear modelos de ML se utilizó la plataforma DR, que permitió tener un panorama amplio del desempeño de diferentes modelos y encontrar entre ellos regresiones de suma utilidad, que en un principio no fueron tenidas en cuenta en la comunidad de ondas gravitacionales. Se ha obtenido que GP con RBK es el modelo de ML que mejor modelo sustituto otorgó de los provenientes de DR, con un margen considerable sobre los demás. Cabe notar que logra superar ampliamente el error del modelo subyacente para todo el rango de parámetros y por lo tanto se logra construir un modelo Surrogate con una regresión de ML sin pérdida de precisión.

En cuanto a modelos de ML obtenidos con DR, el mejor modelo obtenido en cuanto a menor error relativo máximo en el espacio de parámetros es el construido utilizando para todas las regresiones GP con RBK, con el valor de $3,52 \times 10^{-8}$. El modelo que mejor resultado otorgó en cuanto a error relativo medio en el espacio de parámetros fue el construido con los mejores modelos de ML para cada regresión, donde la métrica utilizada para elegir cada uno ha sido RMSE sobre 500 datos de validación equiespaciados. El error relativo medio obtenido fue de $2,25 \times 10^{-9}$. El modelo obtenido únicamente con GP y RBK tiene un error relativo medio similar, de $2,54 \times 10^{-9}$, ya que el primero está constituido mayormente por este tipo de regresiones.

También se comprobó que la construcción de un modelo sustituto a través de los mejores modelos para cada regresión con 5-Fold CV no es óptima, debido a la inherente incerteza de esta métrica. A pesar de esto se pudo construir un modelo que mejore al subyacente. A través de esta métrica también se ha podido elegir el mejor modelo de ML para los conjuntos de datos utilizados (GP con RBK), aunque puede que esto no ocurra en todo caso, debido a la inexactitud de la métrica.

Por otro lado, imponer a todas las regresiones de un modelo sustituto el mismo método de regresión no tiene ninguna justificación más que por simplicidad. En particular los datos que se tratan tienen comportamientos similares, por lo cual tomar para todos los conjuntos de datos el modelo de ML que mejor poder predictivo tenga para describirlos a todos no afecta considerablemente al poder predictivo del modelo sustituto. En caso de querer extraer predicciones con la mayor certeza posible, se debe buscar la mejor regresión para cada conjunto de datos. De esa forma se obtendrá el mejor modelo sustituto. Como se sabe, para alguno de los diferentes conjuntos de datos, algunos modelos pueden ser mejor que otros. Por ejemplo esto se nota

en regresiones de amplitud para varios nodos EIM en la zona de ringdown, donde los modelos ganadores son SVM y no GP, que es lo usual en las regresiones tratadas.

También se utilizaron métodos de regresión clásicos para obtener modelos sustitutos. Al realizar regresiones globales, la mejor elección de n datos es a partir de la jerarquía obtenida por el algoritmo Greedy para construir la base reducida. Para LS, si se eligen n datos para obtener las regresiones, el grado de polinomio óptimo no es el mayor posible (es decir interpolaciones), sino que es menor a n , debido al fenómeno Runge. Los mejores resultados fueron obtenidos con cantidad de datos entre 22 y 26, con polinomios entre grados 18 y 21, con error relativo máximo de 2×10^{-10} . En cuanto a error relativo medio, los mejores resultados se obtuvieron con 25 y 26 datos, con grados de los polinomios entre 19 y 21, con error relativo medio de 2×10^{-12} .

Al obtener regresiones con polinomios, en caso de no contar con un conjunto de datos para validar los resultados de las diferentes regresiones y querer elegir una regresión a partir de los datos de entrenamiento, se recomienda hacer en el caso que se estime que la métrica utilizada sea comparable a la norma global, para evitar problemas por overfitting. Existen trabajos [16] que muestran casos donde se puede mostrar la comparabilidad para LS. La elección de un modelo es mejor a medida que las dos métricas mencionadas son más comparables. Una alternativa a esta problemática es utilizar métodos que no tiendan a realizar overfitting, como lo pueden ser regresiones locales, debido a la no existencia del fenómeno Runge, como lo puede ser Splines.

En cuanto a Splines, el enfoque usado en la práctica de facto, en general es esperable obtener mejores resultados a medida que se agregan datos. Los mejores resultados otorgaron errores relativos mínimos del orden de $3,1 \times 10^{-6}$ y errores relativos medios de $4,3 \times 10^{-8}$. El problema que puede tener Splines Cúbicos es ser un método de complejidad insuficiente como para poder representar de mejor forma el comportamiento de los datos y otorgar predicciones más precisas. Este es el motivo por el cual Splines ha dado resultados menos precisos que los mejores obtenidos a través de LS.

Con respecto a NN, los mejores modelos obtenidos a partir de este método han sido los de mayor poder predictivo de todos los utilizados en el trabajo, con aproximadamente 26 vecinos y polinomios de grado aproximado a 20.

Se encontró que para un grado de polinomio fijo, los modelos a menor número de vecinos tienen performance mayor, esto quiere decir que a medida que se agregan vecinos se pierde capacidad de predicción en donde se evalúa las regresiones. También se encontró que medida que se quitan vecinos, se supera cada vez más a los métodos globales de regresión (equivalentes a trabajar con la cantidad máxima de vecinos cercanos), donde el caso con menor número de vecinos (manteniendo el grado del polinomio fijo) es el de interpolación. El método con mejor convergencia y menores errores es el obtenido con vecinos cercanos e interpolación como método de regresión.

Para NN con polinomios para realizar las regresiones, se puede ver que se necesita tanto localidad como complejidad del modelo de regresión (mayor grado de polinomio) para obtener los mejores resultados.

Se pudo notar que con regresiones de poca complejidad, como por ejemplo con 2 vecinos cercanos e interpolaciones, se obtienen resultados de 10^{-3} para el máximo del error relativo, y con interpolación para 4 vecinos cercanos 10^{-5} , superando la precisión máxima del modelo subyacente, con resultados similares a los obtenidos con Splines Cúbicos.

A partir de los buenos resultados con NN, se puede probar con nuevos métodos provenientes de este enfoque, utilizando otras métricas para definir los vecinos cercanos y otros tipos de regresión. Uno puede ser eligiendo vecinos cercanos menores a un radio r , donde r es un hiperparámetro del problema. Para cada punto del dominio de donde se tomarán los vecinos cercanos, la cantidad de vecinos estará dada por como el algoritmo Greedy definió la distribución de nodos.

¿Qué hace DR con respecto a vecinos cercanos? DR utiliza un método de auto-tuned K-

Nearest Neighbors Regressor (Euclidean Distance), pero no otorga buenos resultados. Utiliza medias ponderadas de los datos, que es un método de regresión simple y poco sofisticado, incapaz de modelar los datos con la precisión que se realizó con NN a través de polinomios.

Se ha visto que modelos más complejos no implican necesariamente mejores resultados. Una muestra de estos es que DR no provee necesariamente la mejor solución, ya que se ha logrado superar en precisión a los mejores modelos de la plataforma con regresiones no incluidas en DR. También, si se dispone de modelos con precisión similar, pero uno es más costoso de generar (por ejemplo una red neuronal vs. una interpolación), debe ser elegido el más simple para evitar complejidad innecesaria y mayor costo computacional para obtener la regresión en tiempo real (del orden de milisegundos).

En este trabajo se obtuvieron como mejores modelos los obtenidos con regresiones a través de 21 vecinos cercanos e interpolación polinomial, donde el modelo predictivo ha tenido un error relativo máximo de $3,19 \times 10^{-11}$. El mejor modelo en cuanto a error medio a lo largo del espacio de parámetros se ha obtenido con 18 vecinos cercanos e interpolación polinomial, con error de $4,25 \times 10^{-13}$. Se debe tener en cuenta que también se han obtenido buenos resultados utilizando LS para obtener regresiones globales. Utilizando todos los datos y polinomios de grado 20 se han obtenido los mejores resultados.

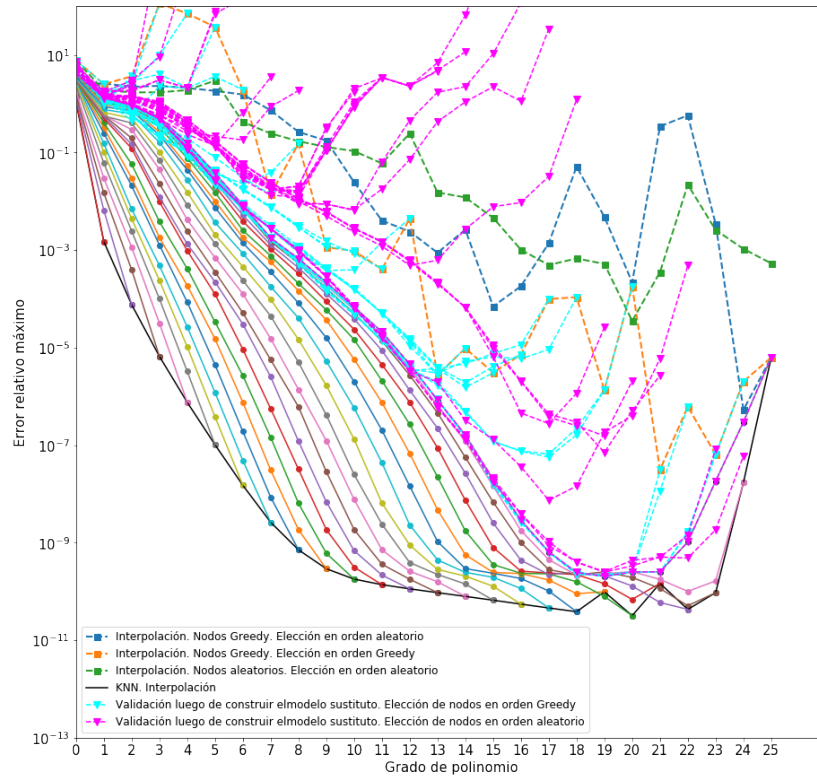


Figura 4.1: Convergencia de los diferentes métodos que utilizan cuadrados mínimos. Para interpolaciones globales se visualiza que el error aumenta para grados de polinomios menores para elección aleatoria de los nodos Greedy, con respecto a la elección en orden Greedy. Para NN y grado de polinomio fijo se obtiene en general mayor precisión a menor cantidad de vecinos cercanos. Para grados altos se observa que las curvas de error crecen, debido a que la cantidad de nodos utilizados es similar al grado del polinomio de LS. El fenómeno se nota en particular para grados mayores, como es de esperar. El modelo que menor error tiene es el de interpolación con 21 vecinos cercanos, superando a las regresiones globales.

A partir de la figura 4.1, se puede realizar un ranking de convergencia al aumentar el grado de polinomio de los diferentes modelos sustitutos obtenidos a través de LS:

1. Interpolación con NN.
2. LS con NN.
3. LS para interpolaciones globales. Nodos Greedy en orden Greedy.
4. LS para interpolaciones globales. Nodos Greedy en orden aleatorio.
5. Interpolación. Nodos Greedy en orden Greedy.
6. Interpolación. Nodos Greedy elegidos de forma aleatoria.
7. Interpolación con nodos aleatorios.

Por último, en la figura 4.2 se puede visualizar el error relativo de los mejores modelos sustitutos obtenidos según diversos enfoques. Se puede visualizar que los modelos obtenidos a través de regresiones simbólicas (utilizando Eureka) con diferentes números de generaciones y SVM no han logrado igualar la precisión del modelo subyacente, provisto por la librería gwsurrogate. Los modelos con GP han otorgado buenos resultados, donde al utilizar MK con $\nu = 1.5, 2.5$ se ha replicado totalmente al modelo subyacente, excepto para $q \approx 1$. Con RB se lo replica en todo el espacio de parámetros. También se visualiza como los modelos obtenidos a través de las mejores regresiones para cada conjunto de datos con RMSE y 5-Fold CV logran dicha replicación. Por el lado de regresiones clásicas se muestra la precisión obtenida con Splines Cúbicos, el mejor modelo obtenido con interpolaciones globales y el mejor modelo con KNN, donde todos logran imitar totalmente al modelo subyacente.

Para observar la contribución al error de las regresiones utilizadas para cada modelo, se ha obtenido el error de un modelo correspondiente a regresiones óptimas, que deja totalmente al descubierto la contribución al error de la elección de la base reducida y los nodos Greedy. Para construirlo, en lugar de realizar regresiones, se han obtenido en cada tiempo empírico los valores de fase y amplitud con el modelo fidedigno. En este caso el error correspondiente a la elección de la base y los tiempos empíricos es a nivel de precisión de máquina: a fines prácticos es casi nulo o directamente nulo. Si contribuyeran, la curva de error relativo se ubicaría por encima del valor de redondeo de la máquina. Por lo tanto se aprecia que la contribución al error proviene de la calidad de las regresiones realizadas en los tiempos empíricos.

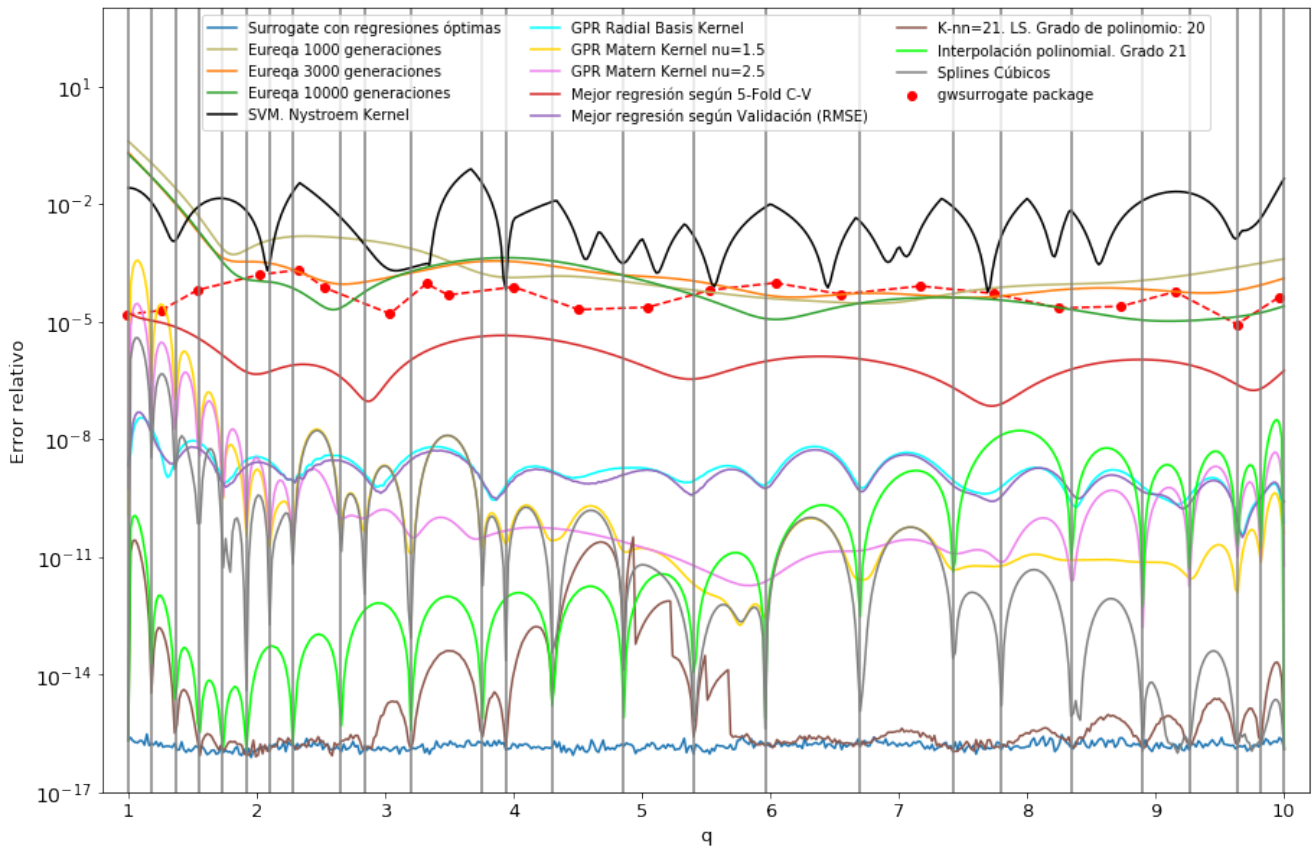


Figura 4.2: Error relativo de los mejores modelos Surrogate obtenidos a través de diferentes métodos. Las líneas verticales denotan la ubicación de los parámetros Greedy obtenidos.

Capítulo 5

Apéndice

5.1. Modelos utilizados por DataRobot

1. Keras Slim Residual Neural Network Regressor using Adaptive Training Schedule (1 Layer: 64 Units)
2. eXtreme Gradient Boosted Trees Regressor with Early Stopping
3. Keras Deep Self-Normalizing Residual Neural Network Regressor using Training Schedule (3 Layers: 256, 128, 64 Units)
4. eXtreme Gradient Boosted Trees Regressor
5. Light Gradient Boosting on ElasticNet Predictions
6. Linear Regression
7. Support Vector Regressor (Linear Kernel)
8. Gradient Boosted Trees Regressor (Least-Squares Loss)
9. Elastic-Net Regressor (mixing $\alpha=0.5$ / Least-Squares Loss) with Unsupervised Learning Features
10. Gaussian Process Regressor with Matern Kernel ($\nu=0.5$)
11. TensorFlow Deep Learning Regressor
12. Gaussian Process Regressor with Matern Kernel ($\nu=1.5$)
13. Ridge Regressor
14. Eureqa Regressor (Quick Search: 1000 Generations)
15. Auto-tuned Stochastic Gradient Descent Regression
16. Ridge Regressor with Binned numeric features
17. RandomForest Regressor (Shallow)
18. Keras Slim Residual Neural Network Regressor using Training Schedule (1 Layer: 64 Units)
19. RandomForest Regressor
20. Keras Wide Residual Neural Network Regressor using Training Schedule (1 Layer: 1536 Units)
21. Gaussian Process Regressor with Radial Basis Function Kernel
22. Vowpal Wabbit Regressor
23. Nystroem Kernel SVM Regressor
24. Eureqa Generalized Additive Model (1000 Generations)
25. Vowpal Wabbit Stagewise Polynomial Regressor
26. Auto-tuned K-Nearest Neighbors Regressor (Euclidean Distance)

27. eXtreme Gradient Boosted Trees Regressor with Early Stopping - Forest (10x)
28. Mean Response Regressor
29. Lasso Regressor
30. ENET Blender
31. Gaussian Process Regressor with Matern Kernel ($\nu=2.5$)
32. Elastic-Net Regressor (mixing $\alpha=0.5$ / Least-Squares Loss)
33. Eureqa Regressor (Long-Running Search: 10000 Generations)
34. Eureqa Generalized Additive Model (10000 Generations)
35. Keras Deep Residual Neural Network Regressor using Training Schedule (2 Layers: 512, 512 Units)
36. eXtreme Gradient Boosted Trees Regressor (learning rate =0.01)
37. Dropout Additive Regression Trees Regressor (15 leaves)
38. Gradient Boosted Greedy Trees Regressor (Least-Squares Loss)
39. Vowpal Wabbit Low Rank Quadratic Regressor
40. Support Vector Regressor (Radial Kernel)
41. Adaboost Regressor
42. Eureqa Regressor (Instant Search: 40 Generations)
43. Eureqa Regressor (Default Search: 3000 Generations)
44. ExtraTrees Regressor
45. eXtreme Gradient Boosted Trees Regressor with Unsupervised Learning Features
46. Ridge Regression
47. Elastic-Net Regressor (mixing $\alpha=0.5$ / Least-Squares Loss) with K-Means Distance Features
48. Keras Deep Residual Neural Network Regressor using Training Schedule (3 Layers: 512, 64, 64 Units)
49. Decision Tree Regressor
50. Eureqa Generalized Additive Model (40 Generations)
51. Generalized Additive Model
52. LightGBM Random Forest Regressor
53. AVG Blender
54. RuleFit Regressor
55. TensorFlow Neural Network Regressor
56. Generalized Additive2 Model
57. Light Gradient Boosted Trees Regressor with Early Stopping

Bibliografía

- [1] Spectral Einstein Code. SpEC. <http://www.black-holes.org/SpEC.html>.
- [2] gwsurrogate. pypi.org/project/gwsurrogate.
- [3] rompy. bitbucket.org/chadgalley/rompy.
- [4] Tor Ovstedal Aanonsen. *Empirical Interpolation with Application to Reduced Basis Approximations*. PhD thesis, Norwegian University of Science and Technology, 2009.
- [5] Harbir Antil, Scott E. Field, Frank Herrmann, Ricardo H. Nochetto, and Manuel Tigli. Two-step greedy algorithm for reduced order quadratures. *CoRR*, abs/1210.0577. arXiv:1210.0577 [cs.NA], 2012.
- [6] J. G. Baker, J. Centrella, D.I. Choi, M. Koppitz, and J. van Meter. Gravitational-wave extraction from an inspiraling configuration of merging black holes. *PRL*, 96:111102. arXiv:gr-qc/0511103v1, 2006.
- [7] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339(9):667 – 672, 2004.
- [8] Peter Binev, Albert Cohen, Wolfgang Dahmen, Ronald A. DeVore, Guergana Petrova, and Przemyslaw Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Analysis*, 43(3):1457–1472, 2011.
- [9] Jonathan Blackman, Scott E. Field, Chad R. Galley, Béla Szilágyi, Mark A. Scheel, Manuel Tiglio, and Daniel A. Hemberger. Fast and accurate prediction of numerical relativity waveforms from binary black hole coalescences using surrogate models. *Phys. Rev. Lett.*, 115:121102, arXiv:1502.07758 [gr-qc], 2015.
- [10] M. Campanelli, C. O. Lousto, P. Marronetti, and Y. Zlochower. Accurate Evolutions of Orbiting Black-Hole Binaries Without Excision. *PRL*, 96:111101, 2006.
- [11] Priscilla Canizares, Scott E. Field, Jonathan R. Gair, Vivien Raymond, Rory Smith, and Manuel Tiglio. Accelerated gravitational-wave parameter estimation with reduced order modeling. *Phys. Rev. Lett.* 114, 071104, arXiv:1404.6284 [gr-qc], 2015.
- [12] Priscilla Canizares, Scott E. Field, Jonathan R. Gair, and Manuel Tiglio. Gravitational wave parameter estimation with compressed likelihood evaluations. *Phys. Rev.*, D87. arXiv:1304.0462 [gr-qc]:124005, 2013.

- [13] Sarah Caudill, Scott E. Field, Chad R. Galley, Frank Herrmann, and Manuel Tiglio. Reduced Basis representations of multi-mode black hole ringdown gravitational waves. *Class. Quant. Grav.*, 29:095016, arXiv:1109.5642 [gr-qc], 2012.
- [14] Joan Centrella, John G. Baker, Bernard J. Kelly, and James R. van Meter. Black-hole binaries, gravitational waves, and numerical relativity. *Rev.Mod.Phys.*, 82:3069, arXiv:1010.5260[gr-qc], 2010.
- [15] Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [16] Albert Cohen, Mark A. Davenport, and Dany Leviatan. On the stability and accuracy of least squares approximations. *Foundations of Computational Mathematics* volume 13, pages 819–834(2013), arXiv:1111.4422 [math.NA], 2018.
- [17] 2001 de Boor C. A Practical Guide to Splines. Springer.
- [18] Scott E. Field, Chad R. Galley, Frank Herrmann, Jan S. Hesthaven, Evan Ochsner, and Manuel Tiglio. Reduced basis catalogs for gravitational wave templates. *Phys. Rev. Lett.*, 106:221102, arXiv:1101.3765[gr-qc], 2011.
- [19] Scott E. Field, Chad R. Galley, Jan S. Hesthaven, Jason Kaye, and Manuel Tiglio. Fast prediction and evaluation of gravitational waveforms using surrogate models. *Phys. Rev. X* 4, 031006 (2014), arXiv:1308.3565 [gr-qc], 2013.
- [20] DataRobot. Enterprise AI. <https://www.datarobot.com>.
- [21] Y. Maday, N. C. Nguyen, A. T. Patera, and S. H. Pau. A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis*, 8:383–404, 2009.
- [22] M. Maggiore. *Gravitational Waves Volume 1: Theory and Experiments*. Oxford University Press, 2008.
- [23] Arnaud Mignan and Marco Broccardo. Neural network applications in earthquake prediction (1994–2019): Meta-analytic and statistical insights on their limitations. *Seismological Research Letters*, 91(4):2330–2342, May arXiv:1910.01178 [cs.NE], 2020.
- [24] Frans Pretorius. Evolution of binary black hole spacetimes. *Phys. Rev. Lett.*, 95:121101, arXiv:gr-qc/0507014, 2005.
- [25] Colin Priest. Data science fails: There’s no such thing as a free lunch. <https://www.datarobot.com/blog/data-science-fails-theres-no-such-thing-as-a-free-lunch>, 2019.
- [26] Olivier Sarbach and Manuel Tiglio. Continuum and Discrete Initial-Boundary-Value Problems and Einstein’s Field Equations. *Living Rev.Rel.*, 15:9, 2012.
- [27] Béla Szilágyi. Key elements of robustness in binary black hole evolutions using spectral methods. *International Journal of Modern Physics D*, 23(07):1430014, arXiv:1405.3693 [gr-qc], 2014.
- [28] C. E. Rasmussen. C. K. I. Williams. Gaussian processes for machine learning. <https://www.GaussianProcess.org/gpml>, 2006.
- [29] David Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, vol.8, 1996.