Optimización de Hiperparámetros de un Sistema de Aprendizaje Supervisado para la Construcción de Ondas Gravitacionales

Atuel E. Villegas A.[1], Franco Cerino[2], Andrés Diaz Pace[3, 4], Manuel Tiglio[2]

1. Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán.
2. Facultad de Matemática, Astronomía, Física y Computación, Universidad Nacional de Córdoba - CONICET.
3. Instituto Superior de Ingeniería de Software Tandil. Unidad Ejecutora de doble dependencia UNCPBA & CONICET.
4. Universidad Nacional del Centro de la Provincia de Buenos Aires.

Resumen

Para la detección e inferencia de parámetros en la detección de ondas gravitaciones se deben generar múltiples funciones de onda, una tarea muy costosa computacionalmente hablando, y que requiere el uso de supercomputadoras. Nuestro objetivo es la construcción de modelos reducidos y sustitutos de alta precisión evaluables en tiempo real en una computadora portátil. Para esto utilizamos un sistema de aprendizaje supervisado, basado en una generalización recientemente introducida del formalismo de bases reducidas usando refinamiento del tipo hp-greedy, la cual resulta en un esquema de partición del espacio de parámetros con una estructura de árbol binario. En este poster tratamos el problema de la optimización de hiperparámetros de este sistema de aprendizaje supervisado utilizando el formalismo Bayesiano, lo que reduce en gran medida el tiempo requerido para la construcción de un modelo óptimo.

Ondas Gravitacionales

Consideramos un sistema binario de agujeros negros sin spin ni carga eléctrica, inicialmente en órbita quasi-circular. Generamos las ondas gravitacionales usando el modelo *NRHybSur3dq8 [1]* para el rango $1 \leq q \leq 10$ donde q es el ratio entre masas.

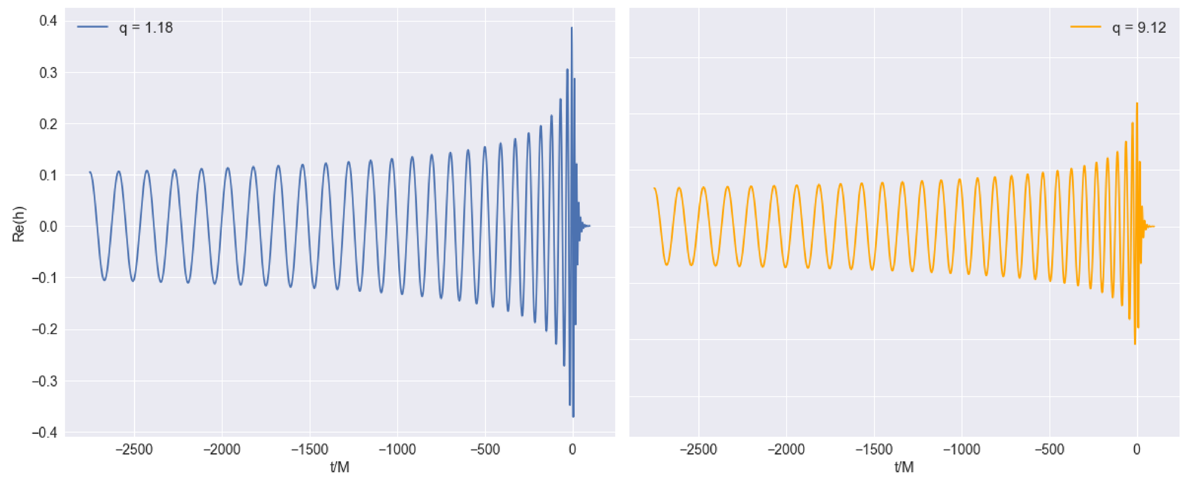

Figura 1. Ondas gravitacionales para dos valores de q.

Bases Reducidas

Es un método que nos permite representar un espacio de funciones $\boldsymbol{F} \coloneqq \{h_\lambda = h_\lambda(t)\}$ a partir de una base constituida por funciones del mismo espacio [2].

Partimos de un conjunto de entrenamiento $\mathbf{K} = \{h_{\lambda_i} : i = 1, \dots, N\}$ con N funciones, y queremos obtener una base ortonormal de n funciones $\{e_i\}_{i=1}^{n}$ de forma que $n \ll N$.

Luego podemos aproximar:

$$h_\lambda \approx P_n h_\lambda = \sum_{i=1}^{n} c_i(\lambda)\, \boldsymbol{e_i}$$

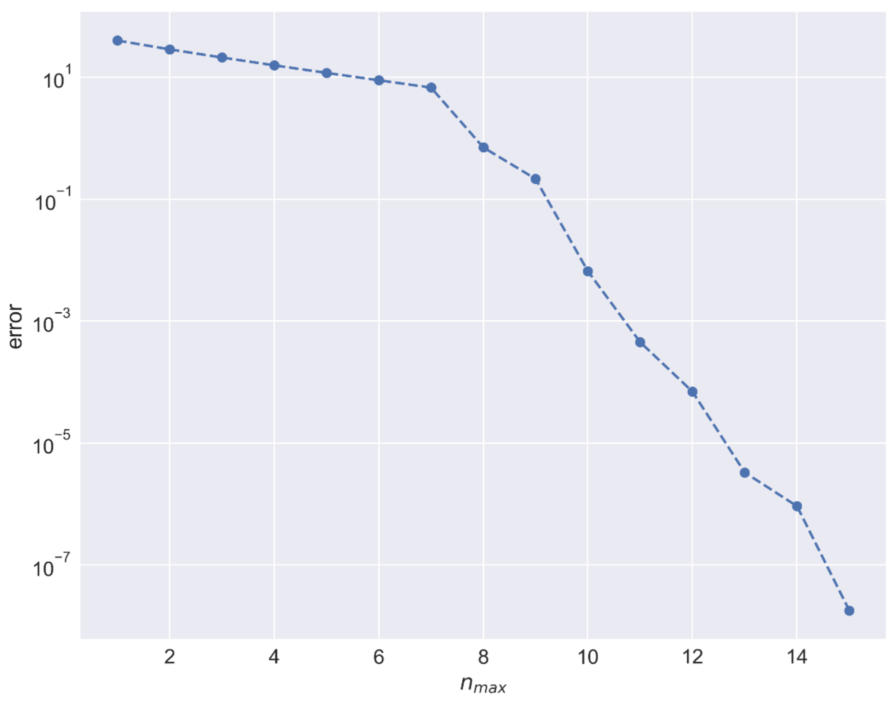

Figura 3. Error de proyección en función del número de elementos de la base reducida.

Para la **construcción de la base** utilizamos un algoritmo de tipo voraz o *greedy*, donde el primer elemento es arbitrario (*semilla)*.
El algoritmo se detiene una vez que se alcanza el número máximo de elementos en la base n_{max}, o cuando el error de representación máximo alcanza la tolerancia definida.

Refinamiento hp-Greedy

Consiste en descomponer el dominio del espacio de parámetros de forma recursiva, dando lugar a la estructura de un árbol binario.

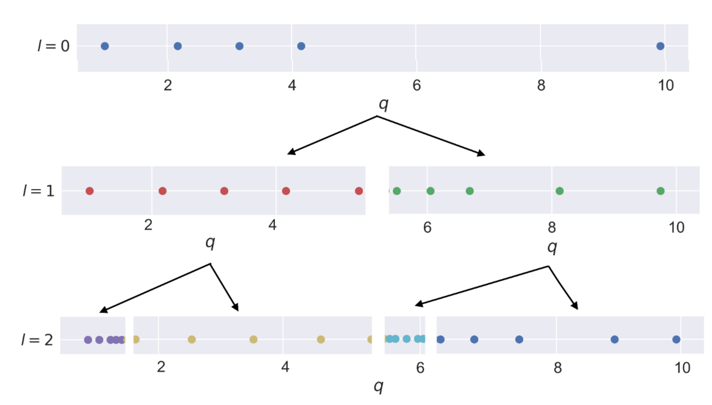

Figura 4. División del espacio de parámetros. Los puntos representan los *greedy points* de la base reducida local de cada dominio.

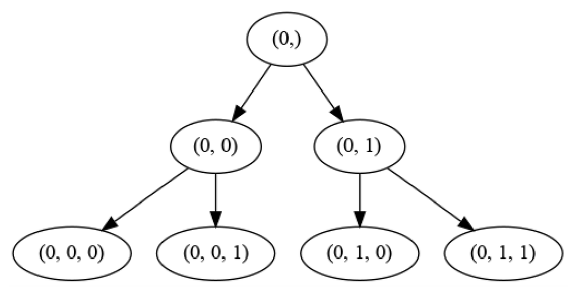

Figura 4. Esquema de la división.

Hiperparámetros

El resultado va a ser sensible a tres hiperparámetros:

La semilla *(seed)*: El parámetro del primer elemento de la base reducida
***l* máximo ($\boldsymbol{l_{max}}$)**: La profundidad máxima del árbol
***n* máximo ($\boldsymbol{n_{max}}$)**: El número máximo de elementos en cada base reducida

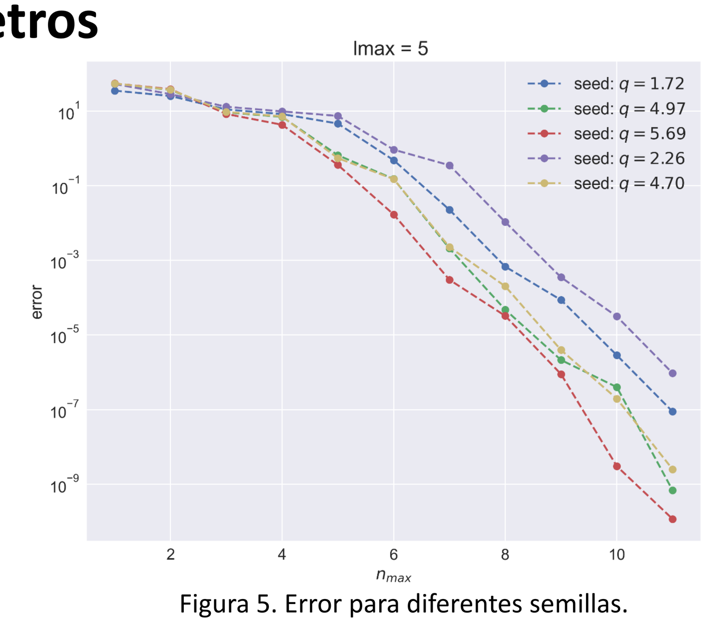

Figura 5. Error para diferentes semillas.

Optimización de Hiperparámetros

Buscamos minimizar una función costosa $f : X \to \Re$

$$\hat{\boldsymbol{x}} = \arg\min_{x \in X} f(\boldsymbol{x})$$

En nuestro caso podemos pensar a X como una caja con tres dimensiones, donde:

$$\boldsymbol{x} = \begin{pmatrix} seed \\ l_{max} \\ n_{max} \end{pmatrix}$$

Y f es una función que nos devuelve el máximo error de representación dado un $\boldsymbol{x}$.

Para encontrar $\hat{\boldsymbol{x}}$ tenemos varias opciones, las más básicas son:
Grid Search: Calcular por fuerza bruta f(x) para todas las combinaciones de $\boldsymbol{x} \in X$.
Random Search: Calcular f(x) para valores aleatorios de $\boldsymbol{x} \in X$

Optimización Bayesiana

En lugar de hacer una búsqueda a ciegas, queremos que cada elección de **x** sea informada a partir de las evaluaciones anteriores de $f(\boldsymbol{x})$**.**

Sea $y = f(\boldsymbol{x})$ queremos construir un modelo estadístico [3]:

$$P(y|\boldsymbol{x})$$

Que se actualice con cada evaluación de $f(\boldsymbol{x})$.

Luego utilizamos una función de Adquisición para elegir los siguientes puntos a evaluar:

$$EI(x) = E[\max(f_{min} - y, 0)]$$

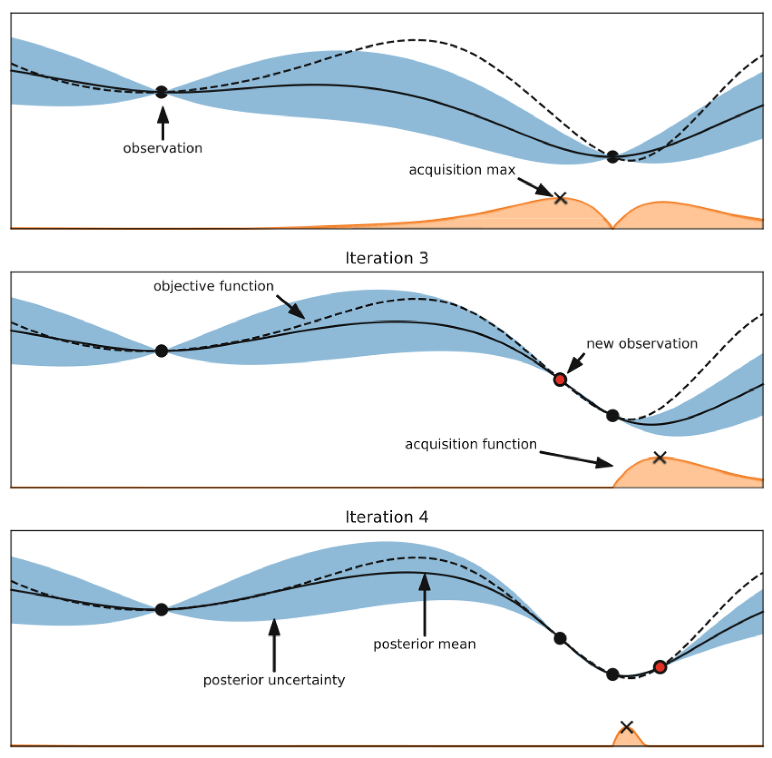

Figura 6. Evolución del modelo estadístico. (fuente [3])

SMBO (*Sequenatial Model-Based Optimization)*

for $i = 1, 2, \dots$ **do**:
  Seleccionar $\boldsymbol{x}_{i+1}$ *optimizando la función de adquisición EI*:
    $x_{i+1} = \arg\max_{\boldsymbol{x}} EI(\boldsymbol{x}, D_i)$
  Evaluar $y_{i+1} = f(\boldsymbol{x}_{i+1})$ (***Paso Costoso***)
  Agregar nuevos datos $D_{i+1} = \mathrm{D_i} \cup (\boldsymbol{x}_{i+1}, y_{i+1})$
  Actualizar Modelo Estadistico $P(y|\boldsymbol{x})$
end **for**

Resultados

Tiempo de ejecución
- **Grid search**: 5 hs.
- **Optimización Bayesiana**: 30 min.

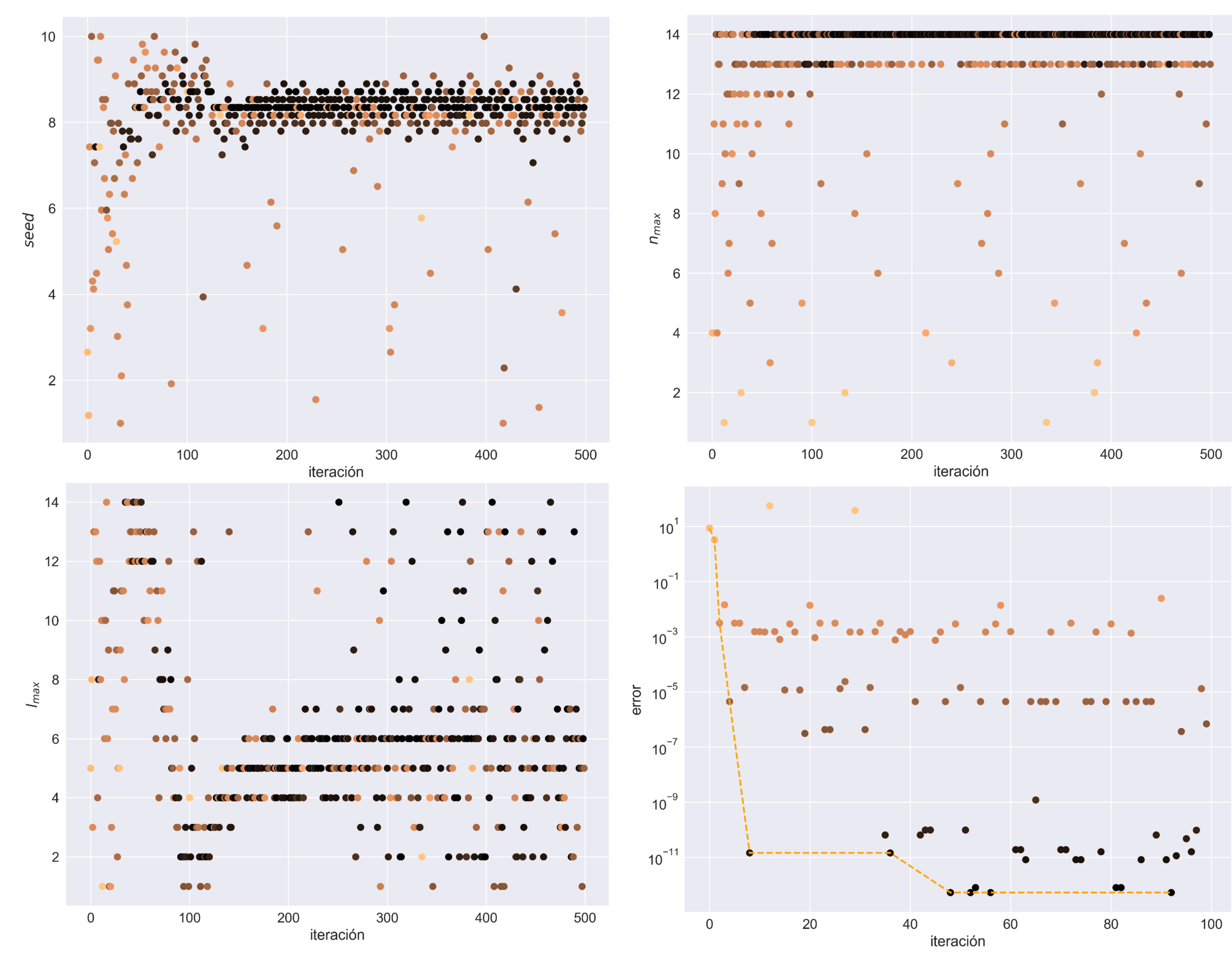

Figura 7. Resultados optimización Bayesiana de hiperparámetros.

Referencias

[1] Surrogate model of hybridized numerical relativity binary black hole waveforms. Vijay Varma, Scott E. Field, Mark A. Scheel, Jonathan Blackman, Lawrence E. Kidder, Harald P. Pfeiffer. Physical Review D (2019).
[2] Tiglio, M. & Villanueva, A. 2021, Reduced Order and Surrogate Models for Gravitational Waves. Living Reviews in Relativity volume 25, Article number: 2 (2022).
[3] Hyperparameter Optimization. In: Hutter, F., Kotthoff, L., Vanschoren, J. (eds) Automated Machine Learning. The Springer Series on Challenges in Machine Learning. Springer (2019).