

Diseño de Agentes Inteligentes

Actividad 4. Búsqueda Local

1: Iván Alejandro López Valenzuela - A01284875

2: Franco Mendoza Muraira - A01383399

3: Alfonso Elizondo Partida - A01285151



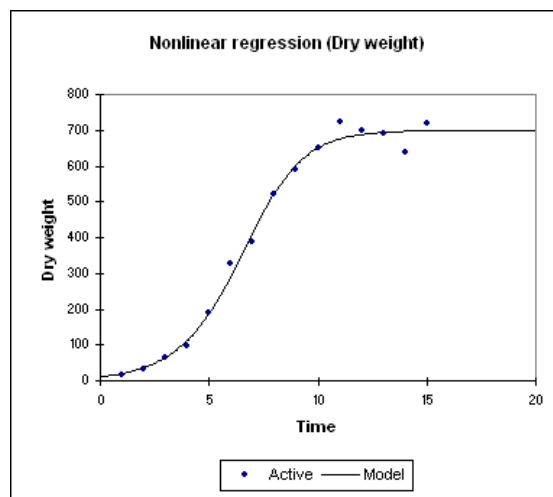
Descripción del problema

Esta actividad consiste en resolver un problema mediante el método de búsqueda local de recocido simulado.

Descripción del Problema de Regresión No Lineal

En estadística, la **regresión no lineal** es un problema de inferencia para un modelo tipo:

$$y = f(x, \theta) + \varepsilon$$



basado en datos multidimensionales \mathbf{x} , \mathbf{y} , donde \mathbf{f} es alguna función no lineal respecto a algunos parámetros desconocidos θ . Como mínimo, se pretende obtener los valores de los parámetros asociados con la mejor curva de ajuste. Con el fin de determinar si el modelo es adecuado, puede ser necesario utilizar conceptos de inferencia estadística tales como intervalos de confianza para los parámetros, así como pruebas de bondad de ajuste. El problema particular para resolver en esta actividad es el de ajustar la función

$$f(x, \theta) = \cos(a x) + b x - c x^2$$

de forma que determinen los valores de los parámetros θ (a , b y c) todos enteros en el rango de $[0, 15]$ que minimicen el máximo valor absoluto del error obtenido por f

$$\arg\min_{\theta} \max |y - f(x, \theta)|$$

sobre los siguientes datos:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
y	1.0	1.0	2.0	4.0	5.0	4.0	4.0	5.0	6.0	5.0

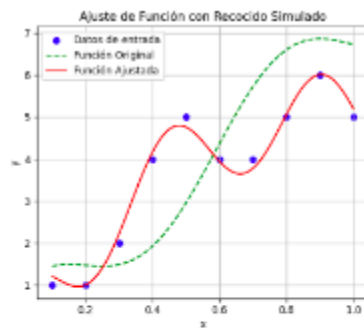
Solución mediante Recocido Simulado

- a) Documenten su solución especificando:
- i) ¿Cómo generarán la solución inicial?
La solución inicial se genera aleatoriamente. Consisten en 3 parámetros: a, b y c.
 - ii) ¿Cómo generarán a los vecinos de la solución actual durante la ejecución del método?
Se utiliza en la función de neighbor. Los vecinos se generan con la suma o resta de un valor aleatorio en cada parámetro.
 - iii) ¿Cómo evaluarán qué tan buena es una solución?, o sea, ¿Cómo calcularán la función objetivo que quieren minimizar?
La función objetivo es el error máximo absoluto entre los valores de la función ajustada y de los valores de entrada.
Se toma el máximo valor absoluto de las diferencias entre la predicción y los datos reales.
- b) Rescriban las funciones de *Python* necesarias para resolver este problema mediante la implementación del método de recocido simulado que encontrarán en el software del curso.
- En el archivo .py adjunto en la carpeta zip.
- En el código se le da al algoritmo el tamaño de paso que va a haber, esto es para los vecinos y cuánto se moverán los parámetros, se proporciona los datos de x y y los cuales se usarán para la función de θ , el α , que es el enfriamiento que se usará para la temperatura, se da la θ inicial, la cual en este caso fueron números random del 0 al 15, y se da la temperatura inicial, la cual es la que irá disminuyendo con las iteraciones que se hagan en el problema. El problema se acaba cuando la temperatura llegaba a ser menor de 0.005 o el error absoluto máximo, que es nuestro costo, llegara a 0, lo cual nunca hizo debido a que es más complejo cuando se trata de funciones debido a lo difícil que es llegar a la exactitud perfecta.
- c) Deben experimentar con diferentes temperaturas iniciales (temperatura) y diferentes razones de reducción de la temperatura (enfriamiento), tratando de encontrar valores adecuados para resolver el problema. Reportar sus conclusiones.

Temperatura inicial: 1

Razón de Enfriamiento: 0.9995

Tamaño del paso: 0.1



Mejor solución encontrada:

a: 13.909874620261535

b: 10.943189527784357

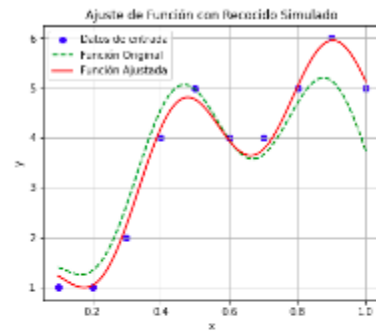
c: 5.969269863019747

Error máximo absoluto:

0.23799020768286372

Temperatura inicial: 1

Alpha: 0.9995



Mejor solución encontrada:

a: 13.89417045015596

b: 11.045656175855495

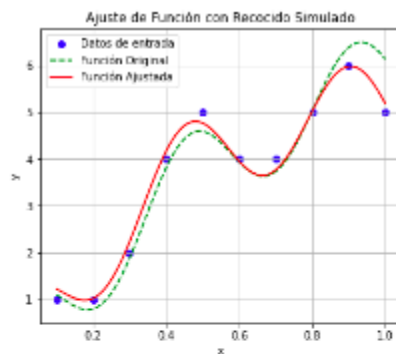
c: 6.157019198467466

Error máximo absoluto:

0.2418844789519934

Temperatura inicial: 1

Alpha: 0.9995



Mejor solución encontrada:

a: 13.879900668292889

b: 10.981237556255618

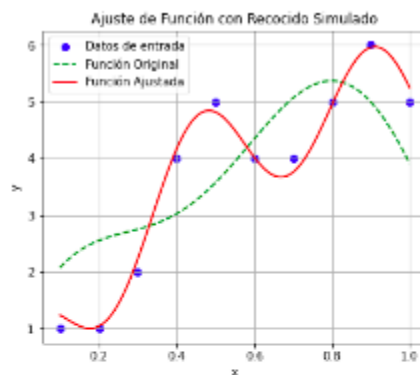
c: 6.044882967306632

Error máximo absoluto:

0.23304004183127391

Temperatura inicial: 1

Alpha: 0.9995



Mejor solución encontrada:

a: 13.787885134740646

b: 11.090150975036934

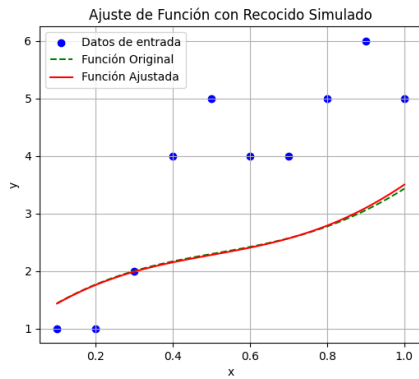
c: 6.187301013912553

Error máximo absoluto:

0.24507302874637027

Temperatura inicial: 2

Alpha: 0.9995



Mejor solución encontrada:

a: 3.280481237665195

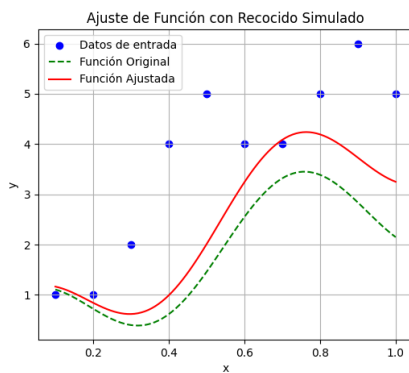
b: 4.8927987130375445

c: 0.40326827150828953

Error máximo absoluto: 2.9075431520144543

Temperatura inicial: 0.5

Alpha: 0.45



Mejor solución encontrada:

a: 8.741117916663951

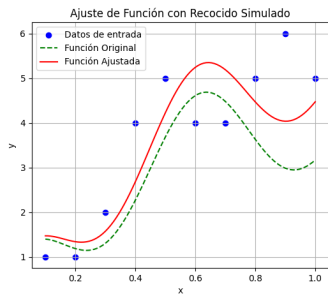
b: 5.35657691455795

c: 1.332290047941828

Error máximo absoluto: 3.008232690596486

Temperatura inicial: 10

Alpha: 0.9



Mejor solución encontrada:

a: 10.327762771172788

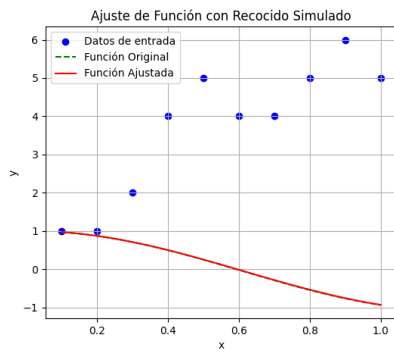
b: 10.051200325747839

c: 4.962630396722451

Error máximo absoluto: 1.965239233360883

Temperatura inicial: 2

Alpha: 0.9



Mejor solución encontrada:

a: 2.623049962436484

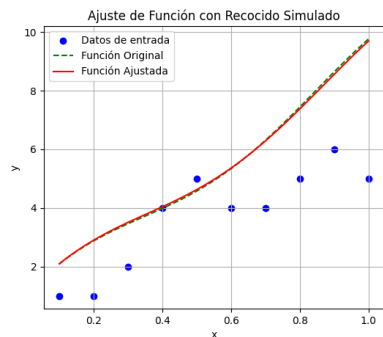
b: 0.04560668473017024

c: 0.10844241194298132


Error máximo absoluto: 6.757109458978191

Temperatura inicial: 2

Alpha: 0.5



Mejor solución encontrada:



a: 5.115859368781714
b: 12.533229820454675
c: 3.229173590362092
Error máximo absoluto: 4.696668660375504
Temperatura inicial: 0.6
Alpha: 0.01

Conclusiones:

Los errores máximos absolutos llegaban hasta 0.2379, pero en general varía mucho, ya que a veces podía tener errores muy altos que incluso superan el 5.

Las soluciones varían en las diferentes ejecuciones ya que los vecinos se generan aleatoriamente al igual que la probabilidad de aceptación.

Ajustar alpha nos ayuda a buscar diferentes convergencias.

El algoritmo contó con un número de iteraciones alto, el cual en algunos casos, tomaba más tiempo en ejecutar.

En las graficas tambien se puede ver como el recocido simulado si sirve mucho para ajustar las funciones, y en general es un buen método para encontrar soluciones, aunque como mencionado anteriormente, puede tomar muchas iteraciones dependiendo de la temperatura inicial y del alpha, o enfriamiento, que le proporcionen.

También como esperado, mientras más baja fuera la temperatura, los resultados terminaron siendo peores debido a la falta de iteraciones que se necesitaban para llegar a un resultado mejoró a un error menor que el que se veía en estos ejemplos. Todo esto claro que puede ser subjetivo debido a la aleatoriedad que se genera debido a la creación de las thetas, pero en general se puede ver mejores resultados y disminuciones en el error mientras más iteraciones se hagan, que esto sucede teniendo una buena selección de temperatura inicial y de alpha, ya que escogiendo alphas altos puede romper el código o dar malos resultados, y dar temperaturas malas muy altas o muy bajas puede dar malos resultados.

