

Diseño de Agentes Inteligentes

Actividad 4. Búsqueda Local



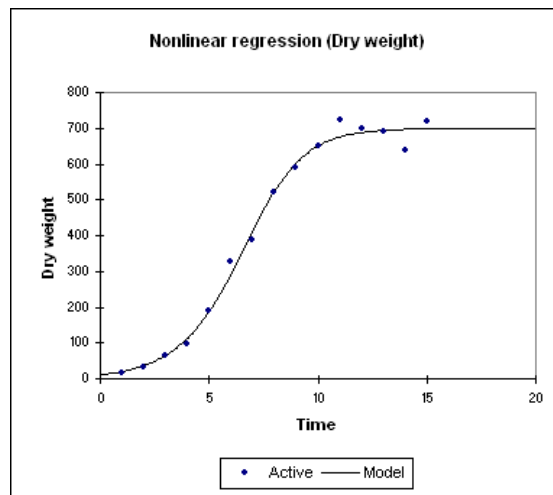
Descripción del problema

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

Miembros del equipo:

- 1: Ricardo Kaleb Flores Alfonso A01198716
- 2: Sebastián Miramontes Soto A01285296
- 3: Raúl Correa Ocañas A01722401

a) Documenten su solución especificando:

i) ¿Cómo generarán la solución inicial?

- (1) Inicialmente asignamos un valor aleatorio del 0 al 15 tanto para a, b y c.
- (2) Generamos el espacio de estados, ya sea incrementando o disminuyendo una unidad de los parámetros. Se revisa que se mantengan en el intervalo [0,15].
- (3) Se determina el error máximo de cada estado, el cual tiene la metaheurística de:
$$\text{error_max} = \max(\text{abs}(y - (\cos(ax) + bx - cx^2)))$$
- (4) Se calcula la temperatura.
- (5) Se consigue a un vecino aleatoriamente, al seleccionar una acción de la lista de acciones aleatoriamente.
- (6) Se calcula la diferencia de errores entre el vecino y el actual
- (7) Si el error del vecino es menor que el actual, se toma el actual.
- (8) En caso de no ser menor que el error actual, entonces si la probabilidad de escoger un número uniformemente del (0,1) es menor que la probabilidad calculada a partir de $\exp(-\text{delta } E / \text{temperatura})$, también se selecciona el vecino.
- (9) Se muestra el estado actual, mostrando una primera solución.

ii) ¿Cómo generarán a los vecinos la solución actual durante la ejecución del método?

- (1) Los vecinos son generados a partir de la solución actual, tomando una acción aleatoria de las posibles que pueden ser tomadas en la solución actual.
- (2) Este proceso es repetido hasta que la temperatura llegue a una probabilidad muy baja (aprox. 0.001) de ser alterada.

iii) ¿Cómo evaluarán que tan buena es una solución?, o sea, ¿Cómo calcularán la función objetivo que quieren minimizar?

- (1) Al querer aproximar los valores de x,y a partir de la función y parámetros asignados, se debe disminuir el error entre el valor real y el aproximado respectivo a los parámetros.
- (2)
$$\text{error_max} = \max(\text{abs}(y - (\cos(ax) + bx - cx^2)))$$
- (3) Por ende buscamos minimizar el error de y con f, con respecto a x y a los parámetros.
- (4) Sabemos que una solución es buena, cuando los parámetros dados implican un máximo error que es menor comparada a las posibles acciones alternativas.
- (5) Tomamos el argumento o parámetros que impliquen éste mínimo error absoluto máximo.

b) Reescriban las funciones de Python necesarias para resolver este problema mediante la implementación del método de recocido simulado que encontrarán en el software del curso.

i) def __init__:

- (1) Contiene estado inicial

- (2) Contiene parámetros iniciales
 - ii) def actions:
 - (1) Enlista todas las posibles acciones, en este caso modificaciones, que se pueden aplicar a los parámetros.
 - iii) def results:
 - (1) dado una acción, modificar los valores de los parámetros.
 - (2) Se revisa que estos no salgan del intervalo [0,15]
 - iv) alpha = 0.9
 - (1) Inicializar la variable de alpha para el método
 - v) to = 1; t = to
 - (1) Inicializar variables de temperatura para el método
 - vi) while t > 0.001:
 - (1) Inicializa el ciclo en el cuál se realiza el proceso de Recocido Simulado
 - vii) t = to * math.pow(alpha, step)
 - (1) Fórmula utilizada para los resultados dentro del método.
 - viii) step += 1
 - (1) Aumenta los pasos o iteraciones del proceso
- 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.
- i) Misma temperatura inicial ($t_{\text{cero}} = 1$), diferentes razones de reducción
 - (1) alpha = 0.99999 # Nada efectivo, (14,11,6) 1:42s
 - (2) alpha = 0.9999 # Menos efectivo, (14,11,6) consistentemente, 4.8s.
 - (3) alpha = 0.999 # Muy eficiente, (14,11,6) consistentemente, 0.8s
 - (4) alpha = 0.99 # Mejor 0.9, (14,11,6),(14,12,7), 0.1s.
 - (5) alpha = 0.9 # Nada confiable, sin respuesta concreta, <0.0s.
 - ii) Misma razón de reducción (alpha = 0.99), diferentes t_{cero}
 - (1) $t_{\text{cero}} = 100000$ # Efectivo, (14,11,6), 0.9s.
 - (2) $t_{\text{cero}} = 10000$ # Efectivo, (14,11,6), 0.9s.
 - (3) $t_{\text{cero}} = 1000$ # Efectivo, (14,11,6), 0.9s.
 - (4) $t_{\text{cero}} = 100$ # Efectivo, (14,11,6), 0.5s.
 - (5) $t_{\text{cero}} = 10$ # Efectivo, (14,11,6), 0.4s
 - (6) $t_{\text{cero}} = 1$ # Efectivo, (14,11,6), 0.3s.

Resultados:

Valores de a, b, c para el mínimo error: 14 12 7
Mínimo valor de error: 0.47973917865930016

Iteration: 16 Temperature: 0.20589113209464907
Valores iniciales a, b, c: {'a': 14, 'b': 12, 'c': 7}
Error máximo: 0.47973917865930016

Valores de a, b, c para el mínimo error: 14 11 6
Mínimo valor de error: 0.8038539587324918

Iteration: 32 Temperature: 0.038152042447694615
Valores iniciales a, b, c: {'a': 15, 'b': 11, 'c': 6}
Error máximo: 0.8038539587324918

Valores de a, b, c para el mínimo error: 14 10 5
Mínimo valor de error: 0.4960977456566953

Iteration: 49 Temperature: 0.006362685
Valores iniciales a, b, c: {'a': 14, 'b': 10, 'c': 5}
Error máximo: 0.4960977456566953

Valores de a, b, c para el mínimo error: 14 14 9
Mínimo valor de error: 0.9355658785102507

Iteration: 34 Temperature: 0.03090315438232636
Valores iniciales a, b, c: {'a': 14, 'b': 14, 'c': 9}
Error máximo: 0.9355658785102507

Conclusiones:

Para esta actividad, a nosotros se nos indico el uso de la búsqueda local mediante Recocido Simulado, que es el método en donde se usa la iteración de la heurística y la temperatura, normalmente este método se utiliza para los problemas de distribución o en donde se deben encontrar movimientos, pero en este caso el código era matemático, más específicamente de la regresión no lineal, en este caso nosotros optamos por dividir el código en tres partes, inicialmente activando las bibliotecas a utilizar, después inicializando variables que se tenían que usar tanto en el proceso como en la toma de resultados, después definimos la clase con todas sus funciones y al final pusimos el proceso de la heurística y la muestra de resultados, con esto pudimos notar que este es un método bueno pero no perfecto para este tipo de escenarios, ya que se le complica mucho el proceso para encontrar la solución, y a pesar de que si muestra la respuesta esperada o una muy acercada, también puede llegar a falla por cualquier desviación mínima.

Como conclusión de este trabajo, podemos decir que los programas de inteligencia artificial deben seguir una estructura clara en la que esté todo conectado y le permita correr el código de manera óptima y mostrando los resultados de manera efectiva, puede llegar a ser un proceso tedioso el iniciar de cero, pero una vez que tienes la estructura solo es cuestión de arreglar detalles mínimos para que pueda funcionar correctamente, por otro lado las búsquedas locales son diferentes al resto, ya que mezclan conceptos del mundo real con lo tecnológico, lo cuál podría ser confuso pero es un método que nos podría servir más a futuro.

Nota:

- Para mí, es muy importante entender como representaron y evaluaron las soluciones para el problema, así como su estrategia para generar las nuevas soluciones en el proceso de búsqueda.

CRITERIOS DE EVALUACIÓN:

Los pesos asignados a las actividades para la evaluación de esta actividad son:

- Implementación de código python para el problema: 60%
- Tabla con resultados obtenidos en los experimentos ejecutados para distintas temperaturas iniciales y razones de enfriamiento: 25%
- Documentación, observaciones y conclusiones: 15%

La calificación por documentación depende de la calidad del documento PDF entregado y la documentación interna del código Python con comentarios.