

LA INTUICIÓN, UNA INTRODUCCIÓN A LAS TÉCNICAS DE BÚSQUEDA HEURÍSTICA

M. en C. Raúl López Muñoz

rlopezm1209@alumno.ipn.mx

RESUMEN

Durante nuestra formación se nos enseñan diferentes técnicas y métodos para resolver problemas, pero en muchas ocasiones deberemos enfrentarnos a situaciones en las que los métodos tradicionales suelen ser difíciles de aplicar o directamente por sus características no son viables; cuando esto sucede, cada persona de acuerdo a diferentes variables como su experiencia e intuición emprenderá un camino particular para hallar una solución, el proceso de “hallar” la respuesta será entonces la heurística.

Introducción

En el presente trabajo se expone a manera de introducción los algoritmos heurísticos y algunas de sus aplicaciones.

El problema

En múltiples disciplinas surge la pregunta ¿cuál es la mejor solución?, para resolver esta pregunta se han seguido distintos enfoques, uno de ellos muy conocido es modelar una función y encontrar el mínimo o máximo (dependiendo del problema) a partir de la obtención de su derivada, por ejemplo. Pero que ocurre cuando la función no es derivable o bien es complejo resolver mediante esta estrategia, cuando se dan estas circunstancias una alternativa es el uso de algoritmos heurísticos; es decir aplicar un conjunto de reglas que sirven de guía para hallar una solución.

Una analogía práctica

El proceso de encontrar el máximo de una función es similar a la acción de un alpinista que desea alcanzar el pico de una montaña. El alpinista se encuentra en algún punto dado y debe decidir qué dirección tomar para subir la montaña. En la vida real es probable que pueda ver la cima, pero en un modelo matemático estamos “vendados” por lo que necesitamos elaborar una estrategia, supongamos que podemos dirigirnos al norte, sur, este y oeste; cada que damos un paso un observador nos dirá si estamos más arriba o más abajo, tras evaluar las

direcciones un camino “lógico” sería dirigirnos en la dirección que nos desplace a un punto superior y en este nuevo punto repetir el proceso.

Un ejemplo sencillo con dos variables

Supongamos que queremos encontrar el valor mínimo de la función $f(x, y) = x^2 + y^2$. La descripción anterior sirve para problemas de encontrar el máximo por lo que para pasar de minimizar a maximizar multiplicaremos la función por un signo negativo $f_{convertida}(x, y) = -f(x, y) = -x^2 - y^2$.

Propondremos el valor de desplazamiento como 0.2 y nuestras direcciones serían entonces.

$$\begin{aligned} P_{Norte} &= [x, y + 0.2] \\ P_{Sur} &= [x, y - 0.2] \\ P_{Este} &= [x + 0.2, y] \\ P_{Oeste} &= [x - 0.2, y] \end{aligned}$$

Supongamos que la posición inicial del alpinista es en $x = 2, y = 2.4$

La función $f_{convertida}(2, 2.4) = -9.76$.

Evaluamos las direcciones: $f_{convertidaN}(2, 2.6) = -10.76$, $f_{convertidaS}(2, 2.2) = -8.84$, $f_{convertidaE}(2.2, 2.4) = -10.6$ y $f_{convertidaO}(1.8, 2.4) = -9$

Como el valor más alto es $x = 2, y = 2.2$ el alpinista se dirige hacia esa solución y evalúa las nuevas direcciones posibles. $f_{convertidaN}(2, 2.4) = -9.76$, $f_{convertidaS}(2, 2) = -8$, $f_{convertidaE}(2.2, 2.2) = -9.68$ y $f_{convertidaO}(1.8, 2.2) = -8.08$

Entonces el nuevo punto será $x = 2, y = 2$. Al evaluar las nuevas soluciones observaremos un fenómeno particular. $f_{convertidaN}(2, 2.2) = -8.84$, $f_{convertidaS}(2, 1.8) = -7.24$, $f_{convertidaE}(2.2, 2) = -8.84$ y $f_{convertidaO}(1.8, 2) = -7.24$

En este caso las posiciones $x = 1.8, y = 2$ y $x = 2, y = 1.8$ tienen el mismo valor en la función, por lo que de acuerdo a nuestra analogía se encuentran a la misma altura por lo que la pregunta es ¿Cuál elegir? Una opción válida es escoger de manera aleatoria una de las dos, o bien establecer un orden jerárquico, es decir, cuando empaten norte y sur siempre ve al norte, por ejemplo. El proceso se realiza hasta que ninguna dirección mejore la posición actual. El resultado final fue $x = 0, y = 0$

Iteración	X	y	f(x,y)	11	1	1,2	-2,44
0	2	2,4	-9,76	12	1	1	-2
1	2	2,2	-8,84	13	0,8	1	-1,64
2	2	2	-8	14	0,8	0,8	-1,28
3	1,8	2	-7,24	15	0,6	0,8	-1
4	1,8	1,8	-6,48	16	0,6	0,6	-0,72
5	1,6	1,8	-5,8	17	0,4	0,6	-0,52
6	1,6	1,6	-5,12	18	0,4	0,4	-0,32
7	1,4	1,6	-4,52	19	0,2	0,4	-0,2
8	1,4	1,4	-3,92	20	0,2	0,2	-0,08
9	1,2	1,4	-3,4	21	0	0,2	-0,04
10	1,2	1,2	-2,88	22	0	0	0

El problema del número de direcciones

Supongamos que tenemos el siguiente problema: maximizar $f(x) = -\sum_{i=1}^{20} x_i^2$. Este problema es similar al anterior solo que en lugar de tener solo x,y tendremos $x_1, x_2, x_3, \dots, x_{20}$. Con el algoritmo anterior tendríamos que evaluar 40 direcciones en cada iteración. Una propuesta para evitar esto es que en cada iteración escojamos una dirección aleatoria (algoritmo RMHC [1]) y evaluaremos si estamos más arriba, de estar más arriba actualizaremos nuestra posición, de lo contrario nos quedaremos en la misma posición y buscaremos en otra dirección aleatoria.

Supongamos la posición aleatoria:

$$X = [1,2,3,4,5,6,7,8,9,10,-1,-2,-3,-4,-5,-6,-7,-8,-9,-10]$$

$$f(X) = -770$$

Elegimos una variable al azar y nos desplazamos aleatoriamente un paso hacia arriba o hacia abajo. Digamos que escogemos desplazarnos hacia arriba 0.1 en $x_3 = 3$.

$$X_{test} = [1,2,3.1,4,5,6,7,8,9,10,-1,-2,-3,-4,-5,-6,-7,-8,-9,-10]$$

$$f(X_{test}) = -770.61$$

En esta dirección hemos descendido por lo que probaremos otra, digamos subir en 0.1 en $x_{11} = -1$.

$$X_{test} = [1,2,3,4,5,6,7,8,9,10,-0.9,-2,-3,-4,-5,-6,-7,-8,-9,-10]$$

$$f(X_{test}) = -769.81$$

Esta vez hemos subido en la colina porque $f(X_{test}) = -769.81 > f(X) = -770$.

Como subimos actualizaremos la posición. En la versión anterior antes de decidir actualizar la posición aun faltarían evaluar otras 18 direcciones.

En la imagen se muestran las actualizaciones de la solución y se observa que en algunos casos le tomó múltiples iteraciones, es decir probar diferentes direcciones antes de encontrar una mejor.

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20	f	i
1	2	3	4	5	6	7	8	9	10	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-770	0
1	2	3	4	5	6	7	8	9	10	-1	-2	-3	-4	-4,9	-6	-7	-8	-9	-10	-769,01	1
1	1,9	3	4	5	6	7	8	9	10	-1	-2	-3	-4	-4,9	-6	-7	-8	-9	-10	-768,62	4
0,9	1,9	3	4	5	6	7	8	9	10	-1	-2	-3	-4	-4,9	-6	-7	-8	-9	-10	-768,43	5
0,9	1,9	3	4	4,9	6	7	8	9	10	-1	-2	-3	-4	-4,9	-6	-7	-8	-9	-10	-767,44	6
0,9	1,9	3	4	4,9	6	7	8	9	9,9	-1	-2	-3	-4	-4,9	-6	-7	-8	-9	-10	-765,45	7
0,9	1,9	3	4	4,9	6	7	8	9	9,9	-1	-2	-2,9	-4	-4,9	-6	-7	-8	-9	-10	-764,86	15
0,9	1,8	3	4	4,9	6	7	8	9	9,9	-1	-2	-2,9	-4	-4,9	-6	-7	-8	-9	-10	-764,49	16
0,9	1,8	2,9	4	4,9	6	7	8	9	9,9	-1	-2	-2,9	-4	-4,9	-6	-7	-8	-9	-10	-763,9	19
0,9	1,8	2,9	4	4,9	6	7	8	9	9,9	-1	-2	-2,9	-3,9	-4,9	-6	-7	-8	-9	-10	-763,11	20

Durante el experimento, el algoritmo encontró que la solución final era un vector de ceros en la iteración número 8656. Es decir, que evaluó 8656 soluciones. Con el algoritmo anterior hubiera tenido que evaluar 44000 para llegar a la misma solución. Si bien el proceso en papel puede ser desgastante las computadoras pueden ayudar en realizar todas las sumas/incrementos y evaluaciones que necesitamos en muy poco tiempo. Por ejemplo, para este problema se programó el algoritmo en MATLAB y para 10000 evaluaciones solo tardó 7.525 ms en ejecutarse. Aunque se debe considerar que para tratar de aproximar una mejor respuesta se debe proponer un paso más pequeño y aumentar las evaluaciones, así como evaluar múltiples veces con diferentes puntos iniciales.

Espacios restringidos

Supongamos que queremos encontrar el valor máximo de la función $f(x) = -x_1^2 - x_2^2$ de tal manera que se cumpla $x_1 + x_2 - 5 = 0$, y además que $x_1 - 1 \geq 0$.

Primero, debido a la forma en que trabaja el algoritmo debemos modificar la condición de igualdad ya que un cambio en la dirección x_1 requeriría otro en x_2 para satisfacer la igualdad.

Una estrategia es realizar la siguiente conversión $|x_1 + x_2 - 5| \leq \varepsilon$. Esta conversión dice que mientras el absoluto de la función sea menor que un valor épsilon lo consideramos como igual. Esta forma permite que, aunque realicemos solo cambios en x_1 mientras el valor de la función este dentro de una tolerancia es válido hacer ese cambio/búsqueda.

Una vez que realizamos la conversión modificaremos las desigualdades para que estén de la forma $g(x) \leq 0$.

El problema reescrito sería encontrar el valor máximo de la función $f(x) = -x_1^2 - x_2^2$ de tal manera que se cumpla $g_1(x) = |x_1 + x_2 - 5| - \varepsilon \leq 0$, y además que $g_2(x) = 1 - x_1 \leq 0$. Una forma de trabajar con restricciones es penalizar el valor de la función, es decir cuando no se cumplan las restricciones la función tendrá un valor muy bajo o muy grande dependiendo si se está maximizando o minimizando. Con esta consideración es posible plantear el problema de la siguiente manera:

Encontrar el valor máximo de la función $f(x) = -x_1^2 - x_2^2 + \mu(\max(0, g_1(x)) + \max(0, g_2(x)))$.

La función max devuelve el valor más grande entre dos elementos. Por ejemplo $\max(3, 2) = 3$.

Volviendo al problema supongamos que queremos evaluar el punto $x_1 = 2, x_2 = 1$, con una tolerancia $\varepsilon = 0.01$ y penalizamos con $\mu = -10000$. El resultado de la función sería:

$$\begin{aligned} f(x) &= -2^2 - 1^2 - 10000(\max(0, |2 + 1 - 5| - 0.01) + \max(0, 1 - 1)) \\ f(x) &= -4 - 1 - 10000(\max(0, |-2| - 0.01) + \max(0, 0)) \\ f(x) &= -5 - 10000(\max(0, 2 - 0.01) + 0) \\ f(x) &= -5 - 10000(\max(0, 1.99)) \\ f(x) &= -5 - 10000(1.99) \\ f(x) &= -199905 \end{aligned}$$

Con estos ajustes podemos volver a aplicar el método previamente descrito, pues el problema se ha convertido de uno con restricciones a uno sin restricciones.

Los mínimos y máximos relativos

Existen funciones dónde no solo existe una cima de la montaña, es decir puede ser que la función represente una zona montañosa y en estos casos existe la posibilidad que dependiendo del lugar donde comencemos, si seguimos el proceso descrito hasta el momento, este nos conducirá a la cima de una montaña, pero no necesariamente de la montaña más alta.

Debido a este fenómeno cuando se implementan las heurísticas es conveniente por no decir necesario ejecutar múltiples veces el algoritmo con distintos puntos de inicio para tratar de obtener la mejor solución.

Aplicaciones

Como hemos observado los algoritmos heurísticos sirven para encontrar una solución que corresponde al máximo o mínimo de una función es decir encuentra una solución que en comparación a las otras es menor o mayor, por este motivo las heurísticas son una estrategia popular en el área de optimización en la cual se obtienen o proponen modelos que cuantifican que tan buena es una solución y después se aplican diferentes técnicas para encontrar la mejor. Algunos ejemplos de ellos son el diseño mecánico, distribución de componentes, administración de horarios, planificación de rutas, identificación de parámetros, etc.

A manera de ejemplo en el caso de la robótica se puede hacer uso de las técnicas heurísticas para obtener la cinemática inversa de robots a partir de la cinemática directa. Supongamos un robot RR. La cinemática directa del robot nos dice que la posición del efector final es:

$$\begin{aligned} Y &= r_1 \sin \theta_1 + r_2 \sin(\theta_1 + \theta_2) \\ X &= r_1 \cos \theta_1 + r_2 \cos(\theta_1 + \theta_2) \end{aligned}$$

Si nos dan la posición en la que queremos que se encuentre el efector y debemos encontrar los ángulos podemos proponer una función de error que dependa de los ángulos y minimizarla. Por lo que el problema sería, por ejemplo:

$$\text{Minimizar } f(\theta_1, \theta_2) = |Y_{Deseada} - r_1 \sin \theta_1 + r_2 \sin(\theta_1 + \theta_2)| + |X_{Deseada} - r_1 \cos \theta_1 + r_2 \cos(\theta_1 + \theta_2)|$$

Si bien existen ecuaciones para la cinemática inversa de este tipo de robots el proceso se puede aplicar para cualquier otro tipo de robot en tanto se conozca su cinemática directa.

Comentarios finales

Hasta el momento hemos visto dos técnicas de búsqueda que se apoyan en una serie de pasos que resultan lógicos y que en la práctica han probado ser útiles para resolver ciertos problemas. Pero el mundo de las técnicas de búsqueda es sumamente vasto de tal suerte que se pueden clasificar por muchos parámetros. Por ejemplo, en que están inspirados si es un proceso natural como la migración de la mariposa monarca [2], si emplean una solución como los mostrados en este artículo o con múltiples soluciones que interactúan entre ellas cómo la optimización por enjambre de partículas [3], si son estáticos o adaptativos por ejemplo existen variantes del algoritmo de escalado de colina con mutación aleatoria y otros algoritmos en el que el paso (el desplazamiento en una dirección) se ajusta dependiendo de ciertos criterios.

Referencias

- [1] S. Forrest and M. Mitchell (1993), "Relative building-block fitness and the building-block hypothesis", in Foundations of Genetic Algorithms Ed. Elsevier, , vol. 2, pp. 109 - 126.
- [2] Wang, GG., Deb, S. & Cui, Z (2019). Monarch butterfly optimization. Neural Comput & Applic 31, 1995–2014
- [3] R. Poli, J. Kennedy, and T. Blackwell (2007). "Particle swarm optimization", Swarm Intelligence, vol. 1, no. 1, pp. 33-57.