Supuestos antes de las explicaciones: se conoce el concepto de función

Explicación Algoritmo de Búsqueda Lineal

Imagina que estás en una alberca enorme y muy profunda. Mientras juegas con tus amigos a los carritos de *Hot Wheels* ocurre una tragedia: ¡se te cayó el carro al fondo de la alberca! Ahora estás metido en un problema muy grande porque tu mamá se enojará mucho contigo si pierdes tu juguete. No sabemos con exactitud dónde se localiza el punto más hondo de la alberca. Además, si te sumerges en la alberca no logras ver con claridad todo el piso y no logras localizar el carrito, aunque sabes que muy probablemente se encuentre en el punto más hondo de la alberca ¿Cómo podemos hacerle para encontrar el carrito?

Paso 1: Elegir un lugar de la alberca a partir del cual vas a empezar a buscar.

Paso 2: Nadar en una dirección fija por debajo del agua mientras sientes la superficie del suelo hasta encontrar un lugar que tenga una profundidad mucho mayor a la profundidad del punto en el que iniciaste.

Paso 3: Una vez que estás en ese nuevo lugar de la alberca, eliges otra dirección para nadar en línea recta que tu creas que te lleve a un lugar más hondo y regresar al paso 2.

Si te fijas, cada vez que aplicas el paso 2 y el paso 3, logras llegar a un lugar de la alberca con una mayor profundidad que el lugar en donde te encontrabas anteriormente. Si sigues aplicando estos pasos, eventualmente llegarás a un lugar muy hondo de la alberca y lograrás encontrar el carrito.

Explicación Algoritmo de Región de Confianza

En el mismo contexto que la explicación pasada, un método alternativo podría ser el siguiente.

Paso 1: Elegir un lugar de la alberca a partir del cual vas a buscar.

Paso 2: Tratas de encontrar el punto más hondo, pero no el más hondo de la alberca, sino el más hondo en un área mucho más pequeña.

Paso 3: Una vez que encuentres el punto más hondo en esa pequeña sección de la alberca, muévete hacia él y repite el paso 2.

NOTAS ADICIONALES:

1. La forma del piso de la alberca es convexa
2. En el paso 2 de la segunda explicación, no se minimiza como tal la función, sino el modelo cuadrático que la aproxima en una vecindad definida.

(1.2) Como la función es convexa, es de utilidad recordar el hecho de que punto estacionario $\Longrightarrow$ mínimo global.

Además, es claro que si $f: \mathbb{R}^n \to \mathbb{R}$ es convexa, entonces cuendo restringimos la función a la recta $x_k + \alpha_k p_k$ la función univariada resultante también es convexa. Con lo cual, el problema se resume a encontrar el punto estacionario.

$$f(x) = \frac{1}{2} x^T Q x - b^T x$$

$$f(x_k + \alpha_k p_k) = \frac{1}{2}(x_k + \alpha_k p_k)^T Q (x_k + \alpha_k p_k) - b^T(x_k + x_k p_k)$$

$$= \frac{1}{2}(x_k^T + \alpha_k p_k^T) Q (x_k + \alpha_k p_k) - b^T (x_k + \alpha_k p_k)$$

$$= \frac{1}{2}(x_k^T + \alpha_k p_k^T)[Q x_k + \alpha_k Q p_k] - b^T x_k - \alpha_k b^T p_k$$

$$= \frac{1}{2}[x_k^T Q x_k + \alpha_k x_k^T Q p_k + \alpha_k p_k^T Q x_k + \alpha_k^2 p_k^T Q p_k] - b^T x_k - \alpha_k b^T p_k$$

Sea $g(\alpha_k) = f(x_k + \alpha_k p_k)$, entonces:

$$g(\alpha_k) = \frac{1}{2}[\alpha_k^2 p_k^T Q p_k + \alpha_k (x_k^T Q p_k + p_k^T Q x_k) + x_k^T Q x_k] - \alpha_k b^T p_k -$$

$$g'(\alpha_k) = \frac{1}{2}[2\alpha_k p_k^T Q p_k + x_k^T Q p_k + p_k^T Q x_k] - b^T p_k$$

<u>Observación</u>: $\nabla f(x) = Qx - b^T \Longrightarrow Qx = \nabla f(x) + b$ y $Q = Q^T$ por conv

$$\therefore g'(\alpha_k) = \frac{1}{2}[2\alpha_k p_k^T Q p_k + (Q x_k)^T p_k + p_k^T Q x_k] - b^T p_k$$

$$= \frac{1}{2}[2\alpha_k p_k^T Q p_k + (\nabla f_k + b)^T p_k + p_k^T (\nabla f_k + b)] - b^T p_k$$

$$= \frac{1}{2}[2\alpha_k p_k^T Q p_k + \nabla f_k^T \cdot p_k + b^T p_k + p_k^T \nabla f_k + p_k^T \cdot b] - b^T p_k$$

$$= \frac{1}{2}\left[2\alpha_k p_k^T Q p_k + \nabla f_k^T p_k + p_k^T \nabla f_k + 2 b^T p_k\right] - b^T p_k$$

$$= \alpha_k p_k^T Q p_k + \nabla f_k^T p_k$$

$$\therefore\ g'(\alpha_k) = 0 \iff \alpha_k p_k^T Q p_k + \nabla f_k^T p_k = 0$$

$$\iff \alpha_k p_k^T Q p_k = -\nabla f_k^T p_k$$

$$\iff \alpha_k = -\frac{\nabla f_k^T p_k}{p_k^T Q p_k} \quad \blacksquare$$

②

Sea $C = \{ y \in \mathbb{R}^2 \mid y$ es una cámara $\}$ con $\# C = M \geq 1$

$D = \{ x \in \mathbb{R}^2 \mid x$ es un delito registrado en la base de datos $\}$ con $\#$

Dado un delito $y_j \in C$, si quiero saber la distancia a alguna de las cámaras basta con calcular la diferencia $\| y_j - x_i \|_2$ con $j \in \{1, \dots, M\}$, $i \in \{1, \dots, N\}$.

Si quiero saber la distancia de la cámara más cercana al delito y_j entonces:

$$\delta_j = \min_{x_i \in D} \| y_j - x_i \|_2$$

De tal forma que, dado un conjunto de cámaras y un conjunto de delitos nos interesa minimizar la siguiente función

$$\min \sum_{j=1}^{M} \delta_j$$

para obtener la distribución óptima de cámaras.

Note que $\delta_j : \mathbb{R}^N \longrightarrow \mathbb{R}$ con lo cual si definimos

$$f(\vec{x}) = \sum_{j=1}^{M} \delta_j \quad \text{con } \vec{x} \in \mathbb{R}^N.$$

Escaneado con CamScanner

Pseudo código:

```
δ ← arreglo de tamaño N
delitos ← matriz de 2×N

cámaras ← matriz de 2×M
resp ← 0
for j in range(1, 2N):

    δ[j] ← min( delitos , cámaras )

    resp ← resp + δ[j]

return resp;
```

función que calcula la distancia mínima entre la j ésima columna de la matriz delitos y todas las columnas de la matriz cámaras