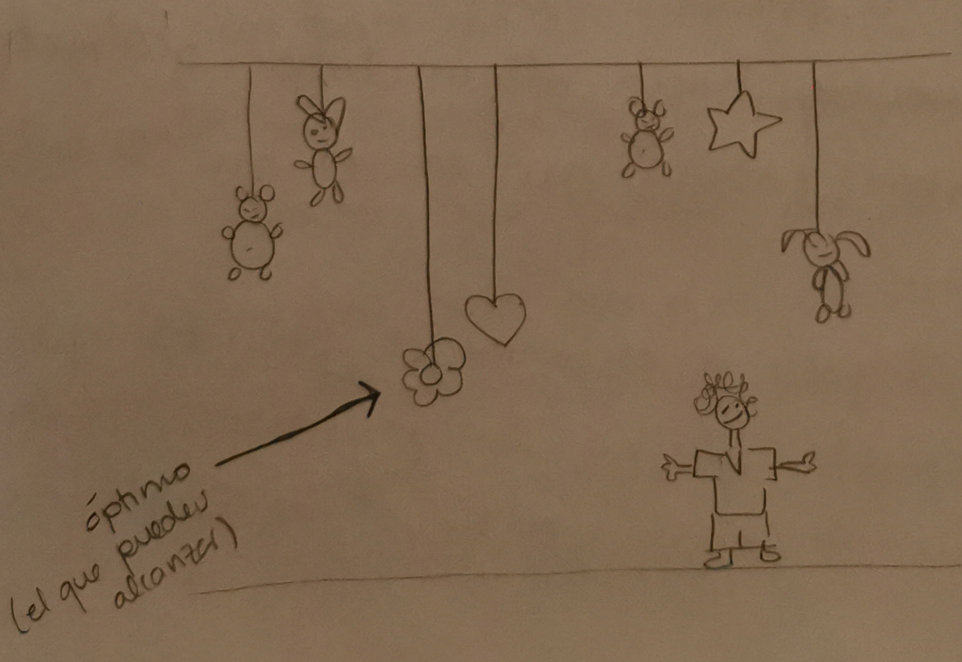
Aimeé León Paredes (1)
142559

1.1.1

Antes de comenzar con estos dos métodos, tienes que saber en qué situación estamos:
Imagínate que estás parado en un salón con peluches colgados desde el techo a diferentes alturas. Eres capaz de alcanzar únicamente uno de esos peluches (el que se encuentra más abajo visto desde el techo).
Para encontrar este peluche deberás, a partir de donde te encuentras parado, ir caminando por todo el lugar de tal forma que puedas lograr encontrar este peluche.



Hay 2 formas de hacerlo:

- Búsqueda Lineal

Desde el lugar en donde estás parado, tendrás que decidir muy bien cada paso que des pues de eso dependerá que puedas ganar/encontrar el peluche.
Para cada paso que des, deberás elegir primero hacia dónde lo darás (es decir, en frente, hacia los lados, etc). Una vez que elijas hacia dónde darás el paso, deberás decidir de qué tamaño darás el paso.
Esto último lo harás mirando hacia la dirección que elegiste y ubicando un peluche que se encuentre más abajo que en el que se encuentra actualmente arriba de ti.
Este proceso/decisión tendrás que hacerlo cada que des un paso. Te detendrás cuando ya no seas capaz de moverte hacia algún lugar que tenga un peluche más abajo que en el que te encuentras actualmente.
Puede que llegues a encontrar el peluche que alcanzas o puede que no lo encuentres, todo dependerá de en dónde inicies el recorrido y

de las decisiones que tomes sobre los pasos que darás.

• Región de confianza

Supongamos ahora que puedes moverte solo dentro de un círculo alrededor de donde te encuentras y no puedes moverte fuera de este círculo.

Nosotros te diremos de qué tamaño es este círculo.

Cuando vayas a decidir hacia dónde y de qué tamaño dar el paso, lo harás mirando hacia ~~arriba~~ y viendo en donde se encuentra el peluche más abajo (sin que debas salir del círculo)

Una vez que lo identifiques, deberás ver si eres capaz de dar ese paso (quizá tus piernas no son tan largas para poder darlo). Si no eres capaz de dar un paso de ese tamaño, entonces haremos el círculo más pequeño y repetiremos el procedimiento hasta que puedas dar un paso.

Esto lo haremos cada que des un nuevo paso y te detendrás hasta que ya no haya un peluche más abajo dentro del círculo en el que te encuentres en ese momento

1.1.2 Demostración

Si tenemos f una cuadrática convexa $f(x) = \frac{1}{2} x^T Q x - b^T x$.

Demuestra que el minimizador de una dimensión sobre la línea $x_k + \alpha p_k$ es:

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

Para empezar, sabemos que como f es convexa, entonces su hessiana Q es definida positiva y simétrica.

También sabemos que: $\nabla f(x) = Qx - b$ y que existe un único x^* mínimo que satisface:

$$Qx - b = 0 \iff Qx = b$$

- Nosotros queremos minimizar $f(x_k + \alpha p_k)$, es decir, minimizar:

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

$\longrightarrow$

Derivamos $f(x_k + \alpha p_k)$ respecto de α:

$$\frac{1}{2}(\nabla f_k^T + p_k^T)Q(x_k + \alpha p_k) + \frac{1}{2}(x_k + \alpha p_k)^T Q(\nabla f_k + p_k) - b^T(\nabla f_k + p_k)$$

◦ Desarrollando y tomando en cuenta que $Q^T = Q$ por simetría, y que $b = Qx_k$:

$$= \frac{1}{2}\nabla f_k^T Q(x_k + \alpha p_k) + \frac{1}{2}p_k^T Q(x_k + \alpha p_k) + \frac{1}{2}x_k^T Q(\nabla f_k + p_k)$$

$$+ \frac{1}{2}\alpha p_k^T Q(\nabla f_k + p_k) - b^T p_k - b^T \nabla f_k$$

$$= \cancel{\frac{1}{2}\nabla f_k^T \overset{b}{Q x_k}} + \frac{1}{2}\nabla f_k^T p_k + \cancel{\frac{1}{2}p_k^T \overset{b}{Q x_k}} + \frac{\alpha}{2}p_k^T Q p_k$$

$$+ \cancel{\frac{1}{2}\overset{b^T}{x_k^T Q}\nabla f_k} + \cancel{\frac{1}{2}\overset{b^T}{x_k^T Q} p_k} + \frac{1}{2}p_k^T \nabla f_k + \frac{\alpha}{2}p_k^T Q p_k$$

$$- \cancel{b^T p_k} - \cancel{b^T \nabla f_k}$$

igualando a cero

$$= \nabla f_k^T p_k + \alpha p_k^T Q p_k = 0$$

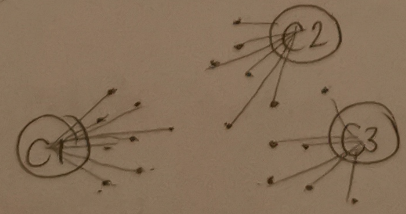
$$\Rightarrow \boxed{\alpha = -\frac{\nabla f_k^T p_k}{p_k^T Q p_k}}$$

1.2

Problema: Contamos con 31,057 crímenes registrados en la ciudad de México. y queremos colocar 8000 cámaras de seguridad. Queremos colocarlas estratégicamente de tal forma que sean capaces de registrar/grabar la mayoría de los delitos que se cometen.

- En la base de datos contamos con la ubicación exacta en coordenadas de cada uno de los crímenes registrados. Nos gustaría que estas 8000 cámaras estén ubicadas de tal forma que se minimice la suma de las distancias de la ubicación de cada crimen a la cámara más cercana.

Una forma de verlo gráficamente sería algo así:



ej con 3 cámaras y 20 crímenes

Por lo tanto, estaríamos diciendo que la función que se optimizará es: la suma de las distancias de cada una de las 31,057 coordenadas a la cámara más cercana.

→ Para empezar, tendríamos que delimitar las coordenadas de la Ciudad de México, pues serán nuestro dominio y es por donde podemos movernos.

Una vez delimitadas/definidas, daremos un "punto inicial" con 8000 entradas, el cual estará dado por 8000 coordenadas aleatorias (dentro del dominio de coordenadas de la CDMX, claro).

Posterior a esto, se definirá la distancia de una coordenada de crimen a la cámara más cercana como el mínimo de las distancias de ese punto a las 8000 cámaras.

El costo justamente estará dado por

$$\sum_{i=1}^{31,057} d(d_i, C_i) \quad , \quad \begin{array}{l} d_i : \text{delito } i \\ C_i : \text{cámara asociada al delito } i \end{array}$$

con $d(d_i, C_i) = \text{mín}\left\{\|d_i - c_j\| \, , \, j \in \{1, 2, \ldots, 8000\}\right\}$

Una vez obtenido este costo "inicial", buscamos minimizar este costo, por lo que queremos movernos sobre las coordenadas dominio de tal forma que la suma de distancias (costo) se haga más pequeña en cada iteración o "paso" que demos:

→ Intentaré programar la función de costo, aunque debemos notar que estamos hablando de una gran cantidad de datos y operaciones y puede conllevar un alto costo computacional.