

---

**Maestría en Computo Estadístico**  
**Optimización**  
**Tarea 2**

3 de mayo de 2021

*Enrique Santibáñez Cortés*

Repositorio de Git: Tarea 1.

1. Dadas las siguientes funciones:

- $f(\vec{x}) = 10(x_2 - x_1^2) + (1 - x_1)^2$
- $f(\vec{x}) = 30 + \sum_{j=1}^2 (x_i^2 - 10 \cos(2\pi x_i))$
- $f(\vec{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
- $f(\vec{x}) = 0,25(x_1^2 + x_2^2) - 0,5(x_1 x_2)$

Resuelva considerando el punto inicial  $\vec{x}^{(0)} = (0, 0)$  y

- a) Gradiente descendente con un tamaño de paso de 0.1.
- b) Método de Newton.
- c) Gradiente descendente con diferencias finitas, considerando un tamaño de paso de 0.1 y una diferencia de  $\pm 0,05$ .

**RESPUESTA:**

Para este ejercicio programamos los algoritmos de optimización en python considerando pseudocódigos vistos en clase, ver el notebook SOLUTIONS\_HOMEWORK.IPYBN. Para facilitar la notación nombremos a las funciones anteriores como  $f_1, f_2, f_3$  y  $f_4$ . Empecemos por encontrar los puntos óptimos de  $f_1$ , para ello primero calculemos el gradiente y su matriz hessiana.

$$\nabla f_1 = \begin{pmatrix} 40x_1(x_2 - x_1^2) - 2(1 - x_1) & 20(x_2 - x_1^2) \end{pmatrix} \quad \text{y} \quad H_{f_1} = \begin{pmatrix} -40x_2 + 120x_1^2 + x & -40x_1 \\ -40x_1 & 20 \end{pmatrix}.$$

Entonces primero calculemos el **algoritmo de Gradiente descendente**, asumiendo como punto inicial  $x^{(0)} = (0, 0)$  tenemos

- 1)  $\nabla f_1(x^{(0)}) = \begin{pmatrix} -2 & 0 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} - 0,01 \nabla f_1(x^{(0)}) = \begin{pmatrix} 0 + ,02 & 0 + 0 \end{pmatrix} = \begin{pmatrix} 0,02 & 0 \end{pmatrix}.$
- 2)  $\nabla f_1(x^{(1)}) = \begin{pmatrix} -1,95968 & -0,008 \end{pmatrix}$   
 $x^{(2)} = \begin{pmatrix} 0,02 + 0,0195968 & 0 + 0,00008 \end{pmatrix} = \begin{pmatrix} 0,0395968 & 0,00008 \end{pmatrix}.$
- 3)  $\nabla f_1(x^{(2)}) = \begin{pmatrix} -1,91844975 & -0,02975813 \end{pmatrix}$   
 $x^{(3)} = \begin{pmatrix} 0,0395968 + 0,0191844975 & 0,00008 + 0,0002975813 \end{pmatrix} = \begin{pmatrix} 0,058781297 & 0,000377581314 \end{pmatrix}.$
- 4)  $\nabla f_1(x^{(3)}) = \begin{pmatrix} -1,87520105 & -0,06155319 \end{pmatrix}$   
 $x^{(4)} = \begin{pmatrix} 0,0587812 + 0,0187520105 & 0,0003775 + 0,0006155319 \end{pmatrix} = \begin{pmatrix} 0,0775333079 & 0,000993113 \end{pmatrix}.$
- 5)  $\nabla f_1(x^{(4)}) = \begin{pmatrix} -1,82936997 & -0,10036601 \end{pmatrix}$   
 $x^{(5)} = \begin{pmatrix} 0,0775333 + 0,0182936997 & 0,00099311 + 0,0010036601 \end{pmatrix} = \begin{pmatrix} 0,095827007 & 0,00199677335 \end{pmatrix}.$
- 6) Realizando algoritmo, converge en la iteración 566 con el punto  $x^{(566)} = \begin{pmatrix} 0,96569466 & 0,93115914 \end{pmatrix}.$

Ahora ocupemos el **método de Newton**,

$$\begin{aligned}
1) \quad \nabla f_1(x^{(0)}) &= \begin{pmatrix} -2 & 0 \\ 0 & 20 \end{pmatrix} \text{ y } H_{f_1} = \begin{pmatrix} 2 & 0 \\ 0 & 20 \end{pmatrix} \\
x^{(1)} &= x^{(0)} - \nabla f_1(x^{(0)}) * H_{f_1}^{-1} = \begin{pmatrix} 0+1 & 0+0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \end{pmatrix}. \\
2) \quad \nabla f_1(x^{(1)}) &= \begin{pmatrix} 40 & -20 \\ -40 & 20 \end{pmatrix} \text{ y } H_{f_1} = \begin{pmatrix} 122 & -40 \\ -40 & 20 \end{pmatrix} \\
x^{(2)} &= \begin{pmatrix} 1+0 & 0+1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}. \\
3) \quad \nabla f_1(x^{(1)}) &= \begin{pmatrix} -8,8817842 - e15 & -4,4408921 - e15 \end{pmatrix} \text{ y } H_{f_1} = \begin{pmatrix} 82 & -40 \\ -40 & 20 \end{pmatrix} \\
x^{(2)} &= \begin{pmatrix} 1+0 & 1+0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}.
\end{aligned}$$

3) Con este algoritmo converge en la tercera iteración, con el punto  $\mathbf{x}^{(3)} = \begin{pmatrix} 1 & 1 \end{pmatrix}$ .

Ahora utilicemos el **Gradiente descendente con diferencias finitas** considerando a  $\epsilon = 0,05$

$$\begin{aligned}
1) \quad \nabla f_1(x^{(0)}) &= \begin{pmatrix} -1,94875 & 0,5 \end{pmatrix} \\
x^{(1)} &= x^{(0)} - 0,01 \nabla f_1(x^{(0)}) = \begin{pmatrix} 0+0,0194875 & 0-0,005 \end{pmatrix} = \begin{pmatrix} 0,0194875 & -0,005 \end{pmatrix}. \\
2) \quad \nabla f_1(x^{(1)}) &= \begin{pmatrix} -1,89749344 & 0,39240475 \end{pmatrix} \\
x^{(2)} &= \begin{pmatrix} 0,0194875 + 0,0189749344 & -0,005 - 0,0039240475 \end{pmatrix} = \begin{pmatrix} 0,0384624 & -0,0089240 \end{pmatrix}. \\
3) \quad \nabla f_1(x^{(2)}) &= \begin{pmatrix} -1,83861115 & 0,29193187 \end{pmatrix} \\
x^{(3)} &= \begin{pmatrix} 0,0384624 + 0,0183861115 & -0,0089240 - 0,0029193187 \end{pmatrix} = \begin{pmatrix} 0,05684854 & -0,0118433662 \end{pmatrix}. \\
4) \quad \nabla f_1(x^{(3)}) &= \begin{pmatrix} -1,77354946 & 0,19849753 \end{pmatrix} \\
x^{(4)} &= \begin{pmatrix} 0,05684854 + 0,0177354946 & -0,0118433662 - 0,0019849753 \end{pmatrix} = \begin{pmatrix} 0,07458404 & -0,0138283 \end{pmatrix}. \\
5) \quad \nabla f_1(x^{(4)}) &= \begin{pmatrix} -1,70375611 & 0,11217759 \end{pmatrix} \\
x^{(5)} &= \begin{pmatrix} 0,07458404 + 0,0170375611 & -0,0138283 - 0,0011217759 \end{pmatrix} = \begin{pmatrix} 0,091621601 & -0,01495011 \end{pmatrix}. \\
6) \quad \text{Realizando algoritmo, converge en la iteración 137 con el punto } x^{(137)} &= \begin{pmatrix} 0,46611116 & 0,19110887 \end{pmatrix}.
\end{aligned}$$

Considerando los tres puntos optimos encontrados utilizando los distintos métodos,

método	número de iteraciones	$x^{optimo}$	$f_1(x^{optimo})$
GD	566	$\begin{pmatrix} 0,96569466 & 0,93115914 \end{pmatrix}$	0.00119
NT	3	$\begin{pmatrix} 1 & 1 \end{pmatrix}$	0
GD finite	137	$\begin{pmatrix} 0,46611116 & 0,19110887 \end{pmatrix}$	0.2918

Cuadro 1: Comparación de los óptimos encontrados de  $f_1$ .

**Observando el Cuadro (??) podemos observar que el metodo de Newton fue el más eficiente para encontrar el óptimo para la función  $f_1$ .**

Análogamente a lo realizado con la función uno lo hacemos con la función dos. Entonces primero calculemos el gradiente y su matriz hessiana de  $f_2$

$$\nabla f_2 = \begin{pmatrix} 2x_1 + 20\pi \sin(2\pi x_1) & 2x_2 + 20\pi \sin(2\pi x_2) \end{pmatrix} \text{ y } H_{f_2} = \begin{pmatrix} 2 + 40\pi^2 \sin(2\pi x_1) & 0 \\ 0 & 2 + 40\pi^2 \sin(2\pi x_2) \end{pmatrix}.$$

Entonces primero calculemos el **algoritmo de Gradiente descendente**, asumiento como punto inicial  $x^{(0)} = (0, 0)$  tenemos

- 1)  $\nabla f_1(x^{(0)}) = \begin{pmatrix} 0 & 0 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} = \begin{pmatrix} 0,0 & 0 \end{pmatrix}$ .
- 2) Es decir, como el gradiente en el punto inicial ya es igual a cero no se mueve el punto, por lo que se encuentra el punto óptimo local.

Ahora ocupemos el **método de Newton**,

- 1)  $\nabla f_1(x^{(0)}) = \begin{pmatrix} 0 & 0 \end{pmatrix}$  y  $H_{f_1} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} - \nabla f_1(x^{(0)}) * H_{f_1}^{-1} = x^{(0)} = \begin{pmatrix} 0 & 0 \end{pmatrix}$ .
- 2) Es decir, como el gradiente en el punto inicial ya es igual a cero no se mueve el punto, por lo que se encuentra el punto óptimo local.

Ahora utilicemos el **Gradiente descendente con diferencias finitas** considerando a  $\epsilon = 0,05$

- 1)  $\nabla f_2(x^{(0)}) = \begin{pmatrix} 0,49193484 & 0,49193484 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} - 0,01 \nabla f_1(x^{(0)}) = \begin{pmatrix} 0 - 0,0049193484 & 0 - 0,0049193484 \end{pmatrix} = \begin{pmatrix} -0,0049193484 & -0,0049193484 \end{pmatrix}$ .
- 2)  $\nabla f_2(x^{(1)}) = \begin{pmatrix} 0,39570972 & 0,39570972 \end{pmatrix}$   
 $x^{(2)} = \begin{pmatrix} -0,0049193484 - 0,39570972 & -0,0049193484 - 0,39570972 \end{pmatrix} = \begin{pmatrix} -0,00887644 & -0,00887644 \end{pmatrix}$ .
- 3)  $\nabla f_2(x^{(2)}) = \begin{pmatrix} 0,31802947 & 0,31802947 \end{pmatrix}$   
 $x^{(3)} = \begin{pmatrix} -0,00887644 - 0,31802947 & -0,00887644 - 0,31802947 \end{pmatrix} = \begin{pmatrix} -0,01205674 & -0,01205674 \end{pmatrix}$ .
- 4)  $\nabla f_2(x^{(3)}) = \begin{pmatrix} 0,25545429 & 0,25545429 \end{pmatrix}$   
 $x^{(4)} = \begin{pmatrix} -0,01205674 - 0,25545429 & -0,01205674 - 0,25545429 \end{pmatrix} = \begin{pmatrix} -0,0146112832 & -0,0146112832 \end{pmatrix}$ .
- 5)  $\nabla f_2(x^{(4)}) = \begin{pmatrix} 0,20511667 & 0,20511667 \end{pmatrix}$   
 $x^{(5)} = \begin{pmatrix} -0,014611 - 0,20511667 & -0,014611 - 0,20511667 \end{pmatrix} = \begin{pmatrix} -0,0166624 & -0,0166624 \end{pmatrix}$ .
- 6) Realizando algoritmo, converge en la iteración 41 con el punto  $x^{(15)} = \begin{pmatrix} -0,02407704 & -0,02407704 \end{pmatrix}$ .

Considerando los tres puntos optimos encontrados utilizando los distintos métodos,

método	número de iteraciones	$x^{optimo}$	$f_1(x^{optimo})$
GD	2	$\begin{pmatrix} 0 & 0 \end{pmatrix}$	10
NT	2	$\begin{pmatrix} 0 & 0 \end{pmatrix}$	10
GD finite	41	$\begin{pmatrix} -0,02407 & -0,02407 \end{pmatrix}$	10.22958

Cuadro 2: Comparación de los óptimos encontrados de  $f_2$ .

**Observando el Cuadro (2) podemos notar que el método de Newton y gradiente descendente fueron los más eficientes.**

Análogamente a lo realizado con las funciones uno y dos lo hacemos con la función tres. Entonces primero calculemos el gradiente y su matriz hessiana de  $f_3$

$$\nabla f_3 = \begin{pmatrix} 2(x_1 + 2x_2 - 7) + 4(2x_1 + x_2 - 5) & 4(x_1 + 2x_2 - 7) + 2(2x_1 + x_2 - 5) \end{pmatrix} \quad \text{y} \quad H_{f_3} = \begin{pmatrix} 10 & 8 \\ 8 & 10 \end{pmatrix}.$$

Entonces primero calculemos el **algoritmo de Gradiente descendente**, asumiento como punto inicial  $x^{(0)} = (0, 0)$  tenemos

- 1)  $\nabla f_3(x^{(0)}) = \begin{pmatrix} -34 & -38 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} - 0,01 \nabla f_3(x^{(0)}) = \begin{pmatrix} 0 + ,34 & 0 + ,38 \end{pmatrix} = \begin{pmatrix} 0,34 & 0,38 \end{pmatrix}.$
- 2)  $\nabla f_3(x^{(1)}) = \begin{pmatrix} -27,56 & -31,48 \end{pmatrix}$   
 $x^{(2)} = \begin{pmatrix} 0,34 + ,2756 & 0,38 + ,3148 \end{pmatrix} = \begin{pmatrix} 0,6156 & 0,69480 \end{pmatrix}.$
- 3)  $\nabla f_3(x^{(2)}) = \begin{pmatrix} -22,2856 & -26,1272 \end{pmatrix}$   
 $x^{(3)} = \begin{pmatrix} 0,6156 + ,222856 & 0,69480 + ,261272 \end{pmatrix} = \begin{pmatrix} 0,838456 & 0,956072 \end{pmatrix}.$
- 4)  $\nabla f_1(x^{(3)}) = \begin{pmatrix} -17,966864 & -21,731632 \end{pmatrix}$   
 $x^{(4)} = \begin{pmatrix} 0,838456 + ,17966864 & 0,956072 + ,21731632 \end{pmatrix} = \begin{pmatrix} 1,01812464 & 1,17338832 \end{pmatrix}.$
- 5)  $\nabla f_3(x^{(4)}) = \begin{pmatrix} -14,43164704 & -18,12111968 \end{pmatrix}$   
 $x^{(5)} = \begin{pmatrix} 1,01812464 + ,1443166704 & 1,17338832 + ,181211196 \end{pmatrix} = \begin{pmatrix} 1,1624411104 & 1,3545995168 \end{pmatrix}.$
- 6) Realizando algoritmo, converge en la iteración 223 con el punto  $x^{(223)} = \begin{pmatrix} 1,01105134 & 2,98894866 \end{pmatrix}.$

Ahora ocupemos el **método de Newton**,

- 1)  $\nabla f_3(x^{(0)}) = \begin{pmatrix} -34 & -38 \end{pmatrix}$  y  $H_{f_3} = \begin{pmatrix} 10 & 8 \\ -1 & -3 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} - \nabla f_1(x^{(0)}) * H_{f_1}^{-1} = \begin{pmatrix} 0 + 1 & 0 + 3 \end{pmatrix} = \begin{pmatrix} 1 & 3 \end{pmatrix}.$
- 2)  $\nabla f_1(x^{(1)}) = \begin{pmatrix} -2,48689958e - 14 & -2,30926389e - 14 \end{pmatrix}$  y  $H_{f_1} = \begin{pmatrix} 10 & 8 \\ -1 & -3 \end{pmatrix}$   
 $x^{(2)} = \begin{pmatrix} 1 & 3 \end{pmatrix}.$
- 3) Con este algoritmo converge en la segunda iteración, con el punto  $\mathbf{x}^{(3)} = \begin{pmatrix} 1 & 3 \end{pmatrix}.$

Ahora utilicemos el **Gradiente descendente con diferencias finitas** considerando a  $\epsilon = 0,05$

- 1)  $\nabla f_3(x^{(0)}) = \begin{pmatrix} -1,6875 & -1,8875 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} - 0,01 \nabla f_3(x^{(0)}) = \begin{pmatrix} 0 + 0,016875 & 0 + 0,018875 \end{pmatrix} = \begin{pmatrix} 0,016875 & 0,018875 \end{pmatrix}.$
- 2)  $\nabla f_3(x^{(1)}) = \begin{pmatrix} -1,6715125 & -1,8713125 \end{pmatrix}$   
 $x^{(2)} = \begin{pmatrix} 0,016875 + ,1671512 & 0,018875 + ,18713 \end{pmatrix} = \begin{pmatrix} 0,033590125 & 0,037588125 \end{pmatrix}.$
- 3)  $\nabla f_3(x^{(2)}) = \begin{pmatrix} -1,65566969 & -1,85526989 \end{pmatrix}$   
 $x^{(3)} = \begin{pmatrix} 0,033590125 + 0,01655669 & 0,037588125 + 0,018552698 \end{pmatrix} = \begin{pmatrix} 0,0501468 & 0,05614082 \end{pmatrix}.$
- 4)  $\nabla f_3(x^{(3)}) = \begin{pmatrix} -1,63997026 & -1,83937086 \end{pmatrix}$   
 $x^{(4)} = \begin{pmatrix} 0,0501468 + 0,01639970 & 0,05614082 + 0,0183937 \end{pmatrix} = \begin{pmatrix} 0,0665465 & 0,074534532 \end{pmatrix}.$
- 5)  $\nabla f_1(x^{(4)}) = \begin{pmatrix} -1,62441292 & -1,82361412 \end{pmatrix}$   
 $x^{(5)} = \begin{pmatrix} 0,0665465 + 0,1624412 & 0,074534532 + 0,1823614 \end{pmatrix} = \begin{pmatrix} 0,0827906 & 0,0927706 \end{pmatrix}.$
- 6) Continuando con el algoritmo, este no converge pero tenemos que en la iteración 1000 el punto  $x^{(1000)} = \begin{pmatrix} 1,35320564 & 2,61855018 \end{pmatrix}.$

Observando el Cuadro (??) podemos observar que el método de Newton fue el más eficiente.

método	número de iteraciones	$x^{optimo}$	$f_3(x^{optimo})$
GD	233	$\begin{pmatrix} 1,01105134 & 2,98894866 \end{pmatrix}$	0.0002442
NT	2	$\begin{pmatrix} 1 & 3 \end{pmatrix}$	0
GD finite	1000	$\begin{pmatrix} 1,35320564 & 2,61855018 \end{pmatrix}$	0.2734

Cuadro 3: Comparación de los óptimos encontrados de  $f_3$ .

Por último, análogamente a lo realizado con las funciones anteriores lo hacemos con la función cuatro. Entonces primero calculemos el gradiente y su matriz hessiana de  $f_4$

$$\nabla f_4 = \begin{pmatrix} 0,5x_1 - 0,5x_2 & 0,5x_2 - 0,5x_1 \end{pmatrix} \quad y \quad H_{f_4} = \begin{pmatrix} 0,5 & -0,5 \\ -0,5 & 0,5 \end{pmatrix}.$$

Entonces primero calculemos el **algoritmo de Gradiente descendente**, asumiento como punto inicial  $x^{(0)} = (0,0)$  tenemos

- 1)  $\nabla f_4(x^{(0)}) = \begin{pmatrix} 0 & 0 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} = \begin{pmatrix} 0,0 & 0 \end{pmatrix}.$
- 2) Es decir, como el gradiente en el punto inicial ya es igual a cero no se mueve el punto, por lo que se encuentra el punto óptimo local.

Ahora ocupemos el **método de Newton**,

- 1)  $\nabla f_4(x^{(0)}) = \begin{pmatrix} 0 & 0 \end{pmatrix}$  y  $H_{f_4} = \begin{pmatrix} 0,5 & -0,5 \\ -0,5 & 0,5 \end{pmatrix}$   
 $x^{(1)} = x^{(0)} - \nabla f_4(x^{(0)}) * H_{f_4}^{-1} = x^{(0)} = \begin{pmatrix} 0 & 0 \end{pmatrix}.$
- 2) Es decir, como el gradiente en el punto inicial ya es igual a cero no se mueve el punto, por lo que se encuentra el punto óptimo local.

Ahora utilicemos el **Gradiente descendente con diferencias finitas** considerando a  $\epsilon = 0,05$

- 1)  $\nabla f_4(x^{(0)}) = (0,000625 \quad 0,000625)$ . Es decir, como el gradiente en el punto inicial ya esta muy cercano a cero no se mueve el punto, por lo que se encuentra el punto óptimo local en  $x^{(0)} = \begin{pmatrix} 0 & 0 \end{pmatrix}.$

Considerando los tres puntos optimos encontrados utilizando los distintos métodos,

método	número de iteraciones	$x^{optimo}$	$f_4(x^{optimo})$
GD	1	$\begin{pmatrix} 0 & 0 \end{pmatrix}$	0
NT	1	$\begin{pmatrix} 0 & 0 \end{pmatrix}$	0
GD finite	1	$\begin{pmatrix} 0 & 0 \end{pmatrix}$	0

Cuadro 4: Comparación de los óptimos encontrados de  $f_4$ .

Por último, **esta función considerando los tres métodos tiene los mismos resultados**. Cabe mencionar que la tolerancia ocupada en los algoritmos fue de  $1e^3$ , por lo que esto puede haber influido a los resultados ■.

2. Utilizando la condición de optimalidad de primer orden para problemas de optimización con restricciones, encuentre la solución óptima de los siguientes problemas:

a)  $f(\vec{x}) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$ , sujeto a  $(x_1 - 1)^3 - x_2 \leq 0$  y  $x_1 + x_2 - 2 \leq 0$ .

**RESPUESTA:**

**Definición: 1** Dada un problema de programación no lineal (NLP) considerando el caso general, es decir, incluyendo restricciones de igualdad y desigualdad en las variables de decisión  $x$ :

$$\begin{aligned} \min \quad & \mathbf{J}(\mathbf{x}) \\ \text{s.a.} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ & \mathbf{g}(\mathbf{x}) \leq \mathbf{0}. \end{aligned}$$

Las condiciones necesarias de óptimo para funciones continuamente diferenciables  $J, h, g$  respecto a las variables  $x$  son:

$$\begin{aligned} \nabla_x J(x) + \sum_{j=1}^n \lambda_j \nabla_x h_j(x) + \sum_{j=1}^n \mu_j \nabla_x g_j(x) &= 0 \\ \mathbf{h}_j(\mathbf{x}) &= \mathbf{0} \\ \mathbf{g}_i(\mathbf{x}) &\leq \mathbf{0} \\ \mu_i \mathbf{g}_i(\mathbf{x}) &= \mathbf{0} \\ \mu_i &\geq 0. \end{aligned}$$

Sea  $L(x, \lambda, \mu) = f(x) - \sum_{j=1}^n \lambda_j h_j(x) - \sum_{j=1}^n \mu_j g_j(x)$ . Entonces, la condición de segundo orden satisface

$$d^T \nabla^2 L(x, \lambda, \mu) \geq 0.$$

Entonces considerando la **Definición 2**, tenemos que las condiciones de optimalidad de primer orden son:

$$\begin{aligned} 2(1 - x_1) - 400x_1(x_2 - x_1^2) + 3\mu_1(x_1 - 1)^2 + \mu_2 &= 0 \\ 200(x_2 - x_1^2) - \mu_1 + \mu_2 &= 0 \\ (x_1 - 1)^3 - x_2 \leq 0, \quad \mu_1 [(x_1 - 1)^3 - x_2] &= 0, \quad \mu_1 \geq 0 \\ x_1 + x_2 - 2 \leq 0, \quad \mu_2 [x_1 + x_2 - 2] &= 0, \quad \mu_2 \geq 0. \end{aligned}$$

Examinemos la combinación más sencilla para  $\mu_1 = \mu_2 = 0$  (por simplicidad), esto representaría a que las dos condiciones del problema original están inactivas. Entonces, considerando  $\mu_1 = \mu_2 = 0$  las condiciones de optimalidad de primer orden se reduce a

$$2(1 - x_1) - 400x_1(x_2 - x_1^2) = 0 \tag{1}$$

$$200(x_2 - x_1^2) = 0 \tag{2}$$

$$(x_1 - 1)^3 - x_2 \leq 0 \tag{3}$$

$$x_1 + x_2 - 2 \leq 0. \tag{4}$$

Despejando (2) tenemos que  $x_2 = x_1^2$ . Ahora, ocupando el resultado anterior y sustituyendolo en (1) tenemos que

$$2(1 - x_1) - 400x_1(x_2 - x_1^2) = 0 \Leftrightarrow 2(1 - x_1) - 400x_1(x_1^2 - x_1^2) = 0 \Rightarrow x_1 = 1. \quad (5)$$

Ocupando (5) en (2) tenemos que  $x_2 = 1$ . Ahora comprobemos si se cumplen las otras dos condiciones de optimalidad:

$$\begin{aligned} (x_1 - 1)^3 - x_2 &\leq 0 \Leftrightarrow (1 - 1)^3 - 1 \leq 0 \Leftrightarrow -1 \leq 0 \\ x_1 + x_2 - 2 &\leq 0 \Leftrightarrow 1 + 1 - 2 \leq 0 \Leftrightarrow 0 \leq 0. \end{aligned}$$

Por lo tanto,  $x_1 = x_2 = 1$  y  $\mu_1 = \mu_2 = 0$  **cumple las condiciones de optimalidad**. Ahora para concluir que esta solución es la solución óptima ocupamos la condición de segundo orden. Para ello primero calculemos  $L(x, \lambda, \mu)$ :

$$L(x, \mu) = (1 - x_1)^2 + 100(x_2 - x_1^2) + \mu_1 [(x_1 - 1)^3 - x_2] + \mu_2(x_1 + x_2 - 2). \quad (6)$$

Ahora calculemos la matriz  $\nabla^2 L(x, \lambda, \mu)$ :

$$\nabla^2 L(x, \lambda, \mu) = \begin{pmatrix} \frac{\partial^2 L(x, \lambda, \mu)}{\partial x^2} & \frac{\partial^2 L(x, \lambda, \mu)}{\partial x \partial y} \\ \frac{\partial^2 L(x, \lambda, \mu)}{\partial x \partial y} & \frac{\partial^2 L(x, \lambda, \mu)}{\partial y^2} \end{pmatrix} = \begin{pmatrix} 2 - 400x_2 + 1200x_1^2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}.$$

Ahora evaluamos  $\nabla^2 L(x, \lambda, \mu)$  con la solución encontrada

$$\nabla^2 L(x^*, \lambda, \mu) = \begin{pmatrix} 2 - 400 + 1200 & -400 \\ -400 & 200 \end{pmatrix} = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}.$$

Veamos si la matriz  $\nabla^2 L(x^*, \lambda, \mu)$  es definida positiva, para ello calculemos sus valores propios. El polinomio característico de  $\nabla^2 L(x^*, \lambda, \mu)$  es

$$p(\lambda) = \det(\nabla^2 L(x^*, \lambda, \mu) - \lambda I) = \begin{vmatrix} 802 - \lambda & -400 \\ -400 & 200 - \lambda \end{vmatrix} = \lambda^2 - 10002\lambda + 400.$$

Esto implica que los valores propios de  $\nabla^2 L(x^*, \lambda, \mu)$  sean:

$$\begin{aligned} p(\lambda) &= 0 \\ \lambda^2 - 10002\lambda + 400 &= (\lambda + \sqrt{250601} - 501)(\lambda - \sqrt{250601} - 501) \Rightarrow \\ \lambda_1 &= 501 - \sqrt{250601} \approx 0,39936 \quad \& \quad \lambda_2 = 501 + \sqrt{250601} \approx 1001,60. \end{aligned}$$

Entonces como los valores propios son ambos positivos, podemos concluir que la matriz  $\nabla^2 L(x^*, \lambda, \mu)$  es positiva definida, es decir, cumple que para cualquier vector  $d^T$  (diferente del nulo) se cumple que  $d^T \nabla^2 L(x^*, \lambda, \mu) d \geq 0$ . Es decir, **la solución encontrada también cumple la condición de segundo orden, por lo que podemos concluir que es la solución óptima de este problema.**

- b)  $f(\vec{x}) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$ , sujeto a  $x_1^2 + x_2^2 \leq 2$ .

**RESPUESTA:**

Considerando la **Definición 2**, tenemos que las condiciones de optimalidad de primer orden son:

$$\begin{aligned} 2(1 - x_1) - 400x_1(x_2 - x_1^2) + 2\mu_1 x_1 &= 0 \\ 200(x_2 - x_1^2) + 2\mu_1 x_2 &= 0 \\ x_1 + x_2 - 2 &\leq 0, \quad \mu_1 [x_1 + x_2 - 2] = 0, \quad \mu_2 \geq 0. \end{aligned}$$

Examinemos la combinación más sencilla para  $\mu_1 = 0$  (por simplicidad), esto representaría a que la restricción del problema original están inactivas. Entonces, considerando  $\mu_1 = 0$  las condiciones de optimalidad de primer orden se reduce a

$$2(1 - x_1) - 400x_1(x_2 - x_1^2) = 0 \quad (7)$$

$$200(x_2 - x_1^2) = 0 \quad (8)$$

$$(x_1 - 1)^3 - x_2 \leq 0 \quad (9)$$

$$x_1 + x_2 - 2 \leq 0. \quad (10)$$

Despejando (2) tenemos que  $x_2 = x_1^2$ . Ahora, ocupando el resultado anterior y sustituyendolo en (1) tenemos que

$$2(1 - x_1) - 400x_1(x_2 - x_1^2) = 0 \Leftrightarrow 2(1 - x_1) - 400x_1(x_1^2 - x_1^2) = 0 \Rightarrow x_1 = 1. \quad (11)$$

Ocupando (5) en (2) tenemos que  $x_2 = 1$ . Ahora comprobemos si se cumple la otra condición de optimalidad:

$$x_1 + x_2 - 2 \leq 0 \Leftrightarrow 1 + 1 - 2 \leq 0 \Leftrightarrow 0 \leq 0.$$

Por lo tanto,  $x_1 = x_2 = 1$  y  $\mu_1 = 0$  **cumple las condiciones de optimalidad**. Ahora consideremos el caso en donde  $\mu_1 \neq 0 (\mu > 0)$  (es decir, la restricción esta activa), el problema se reduce a

$$2(1 - x_1) - 400x_1(x_2 - x_1^2) + 2\mu_1x_1 = 0$$

$$200(x_2 - x_1^2) + 2\mu_1x_2 = 0$$

$$(x_1 - 1)^3 - x_2 \leq 0$$

$$x_1 + x_2 - 2 = 0.$$

Este sistema es un poco complicado de resolverlo a mano, por lo cuál utilizamos Wolfram Alpha para resolverlo. Las soluciones del sistema de ecuaciones que arroja el programa son 7, de las cuales 6 incumplen que  $\mu_1 > 0$ . La otra solución es  $x_1 \approx -1,002214, x_1 \approx 0,997780$  y  $\mu_1 \approx 0,66691$ . Entonces tenemos dos soluciones que cumplen las condiciones de optimalidad de primer orden. Evaluando estas soluciones en nuestra función original tenemos que  $f(x_1^{*1}) = 0$  y  $f(x_1^{*2}) = 4,0132$ . **Por lo que podemos concluir que  $x_1 = x_2 = 1$  es la solución óptima del problema** ■.

NOTA: Para este problema no supe si querían que utilizara algún algoritmo para encontrar la solución óptima, o si se tenían que encontrar explorando posibles soluciones y descartando las que inclupían alguna restricción (es decir, hacerlo de manera manual).

3. Dado la descomposición del problema de optimización de la SVM:

$$\min \frac{1}{2} \vec{\alpha}_B^T Q_{BB} \vec{\alpha}_B - (\vec{e}_B^T - \vec{\alpha}_N^T Q_{NB}) \vec{\alpha}_B + \left( \frac{1}{2} \vec{\alpha}_N^T Q_{NN} \vec{\alpha}_N - \vec{e}_N^T \vec{\alpha}_N \right) \quad (12)$$

sujeto a  $\vec{\alpha}_N^T \vec{y}_B = z$ , donde  $z = -\vec{\alpha}_N^T \vec{y}_N$  y  $B$  y  $N$  representan el subconjunto de variables básicas y no básicas. Considere el algoritmo de SMO, que selecciona las variables básicas  $i$  y  $j$ . Demuestre que el gradiente de  $\alpha_i$  de la Ecuación (12) es equivalente a:

$$\nabla_{\alpha_i} = \begin{cases} \sum_{k=1}^n (\alpha_i \alpha_k Q_{i,k} + \alpha_j \alpha_k Q_{j,k}) - 2 & \text{si } y_i = y_j \\ \sum_{k=1}^n (\alpha_i \alpha_k Q_{i,k} + \alpha_j \alpha_k Q_{j,k}) & \text{si } y_i \neq y_j \end{cases}$$



### RESPUESTA:

Consideremos  $\alpha_i$  y  $\alpha_j$  variables a optimizar usando el algoritmo SMO. Primero observamos que el termino de la Ecuación (12) no depende de las variables  $\alpha_i, \alpha_j$  por lo que lo podemos considerar como constante, es decir, el problema se reduce a

$$\min \frac{1}{2} \vec{\alpha}_B^T Q_{BB} \vec{\alpha}_B - (\vec{e}_B^T - \vec{\alpha}_N Q_{NB}) \vec{\alpha}_B + \text{constante}.$$

Ocupemos la siguiente notación:

$$\vec{\alpha}_B^T = (\alpha_1 \quad \cdots \quad \alpha_i \quad \cdots \quad \alpha_j \quad \cdots \quad \alpha_n) \quad \text{y} \quad Q_{BB} = \begin{pmatrix} Q_{11} & Q_{12} & \cdots & Q_{1n} \\ Q_{21} & Q_{22} & \cdots & Q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Q_{n1} & Q_{n2} & \cdots & Q_{nn} \end{pmatrix}.$$

La expresión anterior es equivalente a

$$\frac{1}{2} \vec{\alpha}_B^T Q_{BB} \vec{\alpha}_B - (\vec{e}_B^T - \vec{\alpha}_N Q_{NB}) \vec{\alpha}_B = \sum_{p=1}^n \alpha_p \left( \sum_{k=1}^n \alpha_k Q_{pk} \right) - \sum_{p=1}^n \alpha_p + \sum_{p=1}^n (\vec{\alpha}_N Q_{NB}) \alpha_p \quad (13)$$

Del resultado (13), intentemos desarrollar más la expresión en terminos de  $\alpha_i$  y  $\alpha_j$ . Primero desarrollemos el primer termino:

$$\begin{aligned} \sum_{p=1}^n \alpha_p \left( \sum_{k=1}^n \alpha_k Q_{pk} \right) &= \alpha_i \left( \sum_{k=1}^n \alpha_k Q_{ik} \right) + \alpha_j \left( \sum_{k=1}^n \alpha_k Q_{jk} \right) + \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p \left( \sum_{k=1}^n \alpha_k Q_{pk} \right) \\ &= \alpha_i^2 Q_{ii} + \alpha_i \alpha_j Q_{ij} + \alpha_i \left( \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{ik} \right) \\ &\quad + \alpha_j \alpha_i Q_{ji} + \alpha_j^2 Q_{jj} + \alpha_j \left( \sum_{k=1}^n \alpha_k Q_{jk} \right) \\ &\quad + \alpha_i \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pi} + \alpha_j \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pj} + \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p \left( \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{pk} \right) \end{aligned}$$

Pero como  $\alpha_i$  y  $\alpha_j$  son variables y el resto se puede considerar como constante, entonces se puede simplificar un poco considerando:

$$\begin{aligned} \sum_{p=1}^n \alpha_p \left( \sum_{k=1}^n \alpha_k Q_{pk} \right) &= \alpha_i^2 Q_{ii} + \alpha_i \alpha_j Q_{ij} + \alpha_i \left( \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{ik} \right) + \alpha_j \alpha_i Q_{ji} + \alpha_j^2 Q_{jj} + \alpha_j \left( \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{jk} \right) \\ &\quad + \alpha_i \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pi} + \alpha_j \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pj} + C_1, \end{aligned}$$

donde  $C_1$  es una constante. Ahora desarrollemos el segundo y tercer termino, y simplifiquemos de igualmanera los terminos constantes:

$$\begin{aligned} - \sum_{p=1}^n \alpha_p + \sum_{p=1}^n (\vec{\alpha}_N Q_{NB}) \alpha_p &= -\alpha_i - \alpha_j - \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p + \vec{\alpha}_N Q_{NB} \alpha_i + \vec{\alpha}_N Q_{NB} \alpha_j + \sum_{\substack{p=1 \\ p \neq i, j}}^n (\vec{\alpha}_N Q_{NB}) \alpha_p \\ &= -\alpha_i - \alpha_j + \vec{\alpha}_N Q_{NB} \alpha_i + \vec{\alpha}_N Q_{NB} \alpha_j + C_2, \end{aligned}$$

donde  $C_2$  es constante.

Lo anterior servirá para facilitar los calculos. Consideraremos dos casos:  $y_i = y_j$ ,  $y_i \neq y_j$ . Cuando  $y_i = y_j$ , implica que  $\alpha_i + \alpha_j = k$  y  $\alpha_j = k - \alpha_i$ . Sustituyamos lo anterior en la expresión simplificada, empecemos por el segundo y tercer termino:

$$\begin{aligned} -\sum_{p=1}^n \alpha_p + \sum_{p=1}^n (\vec{\alpha}_N Q_{NB}) \alpha_p &= -\alpha_i - (k - \alpha_i) + \vec{\alpha}_N Q_{NB} \alpha_i + \vec{\alpha}_N Q_{NB} (k - \alpha_i) + C_2 \\ &= -k + \vec{\alpha}_N Q_{NB} k + C_2 = C_3. \end{aligned}$$

Es decir, lo anterior no depende de las variables  $\alpha_i$  ni  $\alpha_j$  por lo que se consideran como constante ( $C_3$ ). Ahora continuemos con el primer termino:

$$\begin{aligned} \sum_{p=1}^n \alpha_p \left( \sum_{k=1}^n \alpha_k Q_{pk} \right) &= \alpha_i^2 Q_{ii} + \alpha_i (k - \alpha_i) Q_{ij} + \alpha_i \left( \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{ik} \right) + (k - \alpha_i) \alpha_i Q_{ji} + (k - \alpha_i)^2 Q_{jj} \\ &\quad + (k - \alpha_i) \left( \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{jk} \right) + \alpha_i \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pi} + (k - \alpha_i) \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pj} + C_1. \end{aligned}$$

De juntanto los resultados anteriores y derivando respecto a  $\alpha_i$  tenemos que el gradiente de  $\alpha_i$  considerando  $y_i = y_j$  es

$$\begin{aligned} \nabla_{\alpha_i} &= \frac{1}{2} \left( 2\alpha_i Q_{ii} + k Q_{ij} - 2\alpha_i Q_{ij} + \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{ik} + k Q_{ji} - 2\alpha_i Q_{ji} - 2(k - \alpha_i) Q_{jj} - \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{jk} + \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pi} \right) \\ &\quad + \frac{1}{2} \left( - \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pj} + 0 \right) \end{aligned}$$

Recordemos que la matriz  $Q_{BB}$  es simétrica y además que  $k = \alpha_i + \alpha_j$ , entonces podemos la expresión anterior es igual a

$$\begin{aligned} \nabla_{\alpha_i} &= \frac{1}{2} \left( 2\alpha_i Q_{ii} + (\alpha_i + \alpha_j) Q_{ij} - 2\alpha_i Q_{ij} + \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{ik} + (\alpha_i + \alpha_j) Q_{ji} - 2\alpha_i Q_{ji} - 2((\alpha_i + \alpha_j) - \alpha_i) Q_{jj} \right) \\ &\quad + \frac{1}{2} \left( - \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{jk} + \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pi} - \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pj} \right) \\ &= \frac{1}{2} \left( 2\alpha_i Q_{ii} + \alpha_i Q_{ij} + \alpha_j Q_{ij} - 2\alpha_i Q_{ij} + \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{ik} + \alpha_i Q_{ji} + \alpha_j Q_{ji} - 2\alpha_i Q_{ji} - 2\alpha_j Q_{jj} \right) \\ &\quad + \frac{1}{2} \left( - \sum_{\substack{k=1 \\ k \neq i, j}}^n \alpha_k Q_{jk} + \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pi} - \sum_{\substack{p=1 \\ p \neq i, j}}^n \alpha_p Q_{pj} \right) \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \left( 2\alpha_i Q_{ii} + 2\alpha_j Q_{ij} - 2\alpha_i Q_{ij} \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} - 2\alpha_j Q_{jj} - \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} + \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pi} - \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pj} \right) \\
&= \frac{1}{2} \left( 2\alpha_i Q_{ii} + 2\alpha_j Q_{ij} - 2\alpha_i Q_{ij} + 2 \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} - 2\alpha_j Q_{jj} - 2 \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} \right) \\
&= \frac{1}{2} \left( 2\alpha_i Q_{ii} + 2\alpha_j Q_{ij} - 2\alpha_i Q_{ij} + 2 \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} - 2\alpha_j Q_{jj} - 2 \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} \right) \\
&= \frac{1}{2} \left( 2 \sum_{k=1}^n \alpha_k Q_{ik} - 2 \sum_{k=1}^n \alpha_k Q_{jk} \right) = \sum_{\mathbf{k}=1}^n (\alpha_{\mathbf{k}} \mathbf{Q}_{i\mathbf{k}} - \alpha_{\mathbf{j}} \mathbf{Q}_{j\mathbf{k}}).
\end{aligned}$$

Ahora continuemos con el caso cuando  $y_i \neq y_j$ , tenemos que  $\alpha_i - \alpha_j = k \Rightarrow \alpha_j = -k + \alpha_i$ . Realizaremos un procedimiento analago al caso anterior, sustituyamos lo anterior en la expresi3n simplificada del inicio, empecemos por el segundo y tercer termino:

$$\begin{aligned}
-\sum_{p=1}^n \alpha_p + \sum_{p=1}^n (\vec{\alpha}_N Q_{NB}) \alpha_p &= -\alpha_i - (-k + \alpha_i) + \vec{\alpha}_N Q_{NB} \alpha_i + \vec{\alpha}_N Q_{NB} (-k + \alpha_i) + C_2 \\
&= -2\alpha_i + k + \vec{\alpha}_N Q_{NB} k + 2\vec{\alpha}_N Q_{NB} \alpha_i + C_2.
\end{aligned}$$

Ahora continuemos con el primer termino:

$$\begin{aligned}
\sum_{p=1}^n \alpha_p \left( \sum_{k=1}^n \alpha_k Q_{pk} \right) &= \alpha_i^2 Q_{ii} + \alpha_i (-k + \alpha_i) Q_{ij} + \alpha_i \left( \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} \right) + (-k + \alpha_i) \alpha_i Q_{ji} + (-k + \alpha_i)^2 Q_{jj} \\
&\quad + (-k + \alpha_i) \left( \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} \right) + \alpha_i \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pi} + (-k + \alpha_i) \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pj} + C_1.
\end{aligned}$$

De juntanto los resultados anteriores y derivando respecto a  $\alpha_i$  tenemos que el gradiente de  $\alpha_i$  considerando  $y_i \neq y_j$  es

$$\begin{aligned}
\nabla_{\hat{\alpha}_i} &= \frac{1}{2} \left( 2\alpha_i Q_{ii} - k Q_{ij} + 2\alpha_i Q_{ij} + \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} - k Q_{ji} + 2\alpha_i Q_{ji} + 2(-k + \alpha_i) Q_{jj} - \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} + \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pi} \right) \\
&\quad + \frac{1}{2} \left( \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pj} \right) - 2
\end{aligned}$$

Recordemos que la matriz  $Q_{BB}$  es simétrica y adem3s que  $k = \alpha_i - \alpha_j$ , entonces podemos la expresi3n

anterior es igual a

$$\begin{aligned}
\nabla_{\hat{\alpha}_i} &= \frac{1}{2} \left( 2\alpha_i Q_{ii} - (\alpha_i - \alpha_j) Q_{ij} + 2\alpha_i Q_{ij} + \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} - (\alpha_i - \alpha_j) Q_{ji} + 2\alpha_i Q_{ji} - 2(-(\alpha_i - \alpha_j) + \alpha_i) Q_{jj} \right) \\
&\quad + \frac{1}{2} \left( \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} + \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pi} + \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pj} \right) - 2 \\
&= \frac{1}{2} \left( 2\alpha_i Q_{ii} - \alpha_i Q_{ij} + \alpha_j Q_{ij} + 2\alpha_i Q_{ij} + \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} - \alpha_i Q_{ji} + \alpha_j Q_{ji} + 2\alpha_i Q_{ji} - 2\alpha_j Q_{jj} \right) \\
&\quad + \frac{1}{2} \left( \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} + \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pi} + \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pj} \right) - 2 \\
&= \frac{1}{2} \left( 2\alpha_i Q_{ii} + 2\alpha_j Q_{ij} + 2\alpha_i Q_{ij} + \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} + 2\alpha_j Q_{jj} + \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} + \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pi} + \sum_{\substack{p=1 \\ p \neq i,j}}^n \alpha_p Q_{pj} \right) - 2 \\
&= \frac{1}{2} \left( 2\alpha_i Q_{ii} + 2\alpha_j Q_{ij} + 2\alpha_i Q_{ij} + 2 \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} + 2\alpha_j Q_{jj} + 2 \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} \right) - 2 \\
&= \frac{1}{2} \left( 2\alpha_i Q_{ii} + 2\alpha_j Q_{ij} + 2\alpha_i Q_{ij} + 2 \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{ik} + 2\alpha_j Q_{jj} + 2 \sum_{\substack{k=1 \\ k \neq i,j}}^n \alpha_k Q_{jk} \right) - 2 \\
&= \frac{1}{2} \left( 2 \sum_{k=1}^n \alpha_k Q_{ik} + 2 \sum_{k=1}^n \alpha_k Q_{jk} \right) - 2 = \sum_{k=1}^n (\alpha_k Q_{ik} + \alpha_j Q_{jk}) - 2.
\end{aligned}$$

Entonces podemos concluir que el gradiente descendente para  $\alpha_i$  es

$$\nabla_{\hat{\alpha}_i} = \begin{cases} \sum_{k=1}^n (\alpha_k Q_{i,k} + \alpha_k Q_{j,k}) - 2 & \text{si } y_i \neq y_j \\ \sum_{k=1}^n (\alpha_k Q_{i,k} - \alpha_k Q_{j,k}) & \text{si } y_i = y_j \end{cases} \quad \blacksquare.$$

4. Formule el problema de regresión con regularización como un problema de programación cuadrática e implemente una función en Python que reciba los datos y determine los coeficientes.

**RESPUESTA:**

Tenemos que el problema de regresión con regularización se puede escribir como el problema de programación cuadrática:

$$\min_{\beta} \quad \|\beta^T X - y\|_2^2 \tag{14}$$

$$s.a \quad \|\beta\| < t. \tag{15}$$

Lo anterior se conoce como regularización de Ivanov, que es equivalente a la regularización de Tikhonov:

$$\min_{\beta} \quad \|\beta^T X - y\|_2^2 + \lambda \|\beta\|.$$

Veamos que la formulación de regresión con regularización (14) es un problema de programación cuadrático. Entonces la suma del cuadrado del error se es equivalente a

$$RSS(b) \propto b^T (X^T X) b - 2y^T X \beta.$$

Entonces, podemos usar a  $Q = 2(X^T X)$  y  $q^T = 2y^T X$  de la definición de un problema de programación cuadrática. Además, para escribir  $A$  se tiene que considerar  $2^p$  restricción de desigualdad. Por ejemplo, cuando  $p = 2$  la restricción  $|\beta_1| + |\beta_2| \leq t$  es equivalente a:

$$\begin{aligned}\beta_1 + \beta_2 &\leq t \\ \beta_1 - \beta_2 &\leq t \\ -\beta_1 + \beta_2 &\leq t \\ -\beta_1 - \beta_2 &\leq t,\end{aligned}$$

Entonces  $A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix}$  y  $b = (t \ t \ t \ t)$ . Cabe resaltar que  $2^p$  puede ser demasiado grande

cuando  $p$  crece, por lo que este plantamiento del problema es muy tardado. Entonces con lo anterior podemos resolver el problema de forma cuadrática con ayuda de python, el cual consiste simplemente en crear las matrices anteriores descritas. Para la función ocupamos las librerías `itertools`, `pandas`, `numpy` y `cvxopt`,

```
import itertools # funcion auxiliar para crear A
import pandas as pd # transformar dataframe.
from cvxopt import matrix, solvers # resolver un problema cuadrático

### funcion auxiliar:

# Crear la matriz A.
def expand_grid(data_dict):
    """Crear la matriz de las combinaciones de A.

    Args:
        data_dict: (p, 2) diccionario con los vectores (-1,1)

    return:
        pd.DataFrame: (p, 2) matriz A.
    """
    rows = itertools.product(*data_dict.values())
    return pd.DataFrame.from_records(rows, columns=data_dict.keys())

### Regression con regularizacion l1.
def regression_lasso_qp(X, y, t):
    """
    Estimacion de beta para una regression con regularizacion l1.

    Args:
        X: (n,p) np.array, los datos del problema
        y: (n) np.array, variable respuesta
        t: parametro t de regularizacion
```

---

```

return:
    sol["x"]: solucion optima encontrada
"""
n, p = X.shape # tamaño de los datos

D = matrix(X.T@X, tc="d") # calculamos la matriz D
q = matrix(2*y.T@X, tc="d") # calculamos el vector q

a = np.array([-1,1]) # calculamos la matriz A
aux = {}
for i in range(p):
    aux["Var{}".format(i)] = a
A = matrix(np.array(expand_grid(aux)), tc="d")

b0 = matrix(np.repeat(t, 2**p).reshape((2**p)), tc="d") # vector b

sol = solvers.qp(D, q, A, b0) # resolvemos el PQ.
return print(sol["x"])

```

Ocupemos los datos de sklearn Boston para probar nuestra función:

```

from sklearn.datasets import load_boston # cargamos los datos
X, y = load_boston(return_X_y=True)

```

```

regression_lasso_qp(X[:,0:2], y, 10) # usamos la funcion

```

# output del ejemplo:

	pcost	dcost	gap	pres	dres
0:	-1.9668e+05	-1.9680e+05	1e+02	2e-17	5e-17
1:	-1.9668e+05	-1.9669e+05	1e+00	1e-16	2e-17
2:	-1.9668e+05	-1.9668e+05	1e-02	2e-16	2e-17

Optimal solution found.

```

[-1.16e+00]
[-9.90e-01]

```