Centro de Investigación en Matemáticas (CIMAT). Unidad Monterrey

Optimización
Tarea 2

Marcelo Alberto Sanchez Zaragoza

2 de mayo de 2021

1. Problema 1

Dadas las siguientes funciones:

- f($\overrightarrow{x}$) = $10(x_2 - x_1^2)^2 + (1 - x_1)^2$
- f($\overrightarrow{x}$) = $30 + \sum_{i=1}^{2}(x_i^2 - 10cos(2\pi x_i))$
- f($\overrightarrow{x}$) = $(x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
- f($\overrightarrow{x}$) = $0.25(x_1^2 + x_2^2)$ - $0.5(x_1 x_2)$

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

a) Gradiente descendente con un tama no de paso de 0.01.

b) Método de Newton.

c) Gradiente descendente con diferencias

finitas, considerando un tamaño de paso de 0.01 y una diferencia de ± 0.05.

Solución

Para encontrar cada de una de las aproximaciones nos apoyamos del lenguaje de programación llamado Python, se anexa el programa.

Inciso a

Para el primer inciso encontramos cada uno de los gradientes de las funciones proporcionadas ya que lo pide el método que nos solicitan, estos gradientes se ocuparon para encontrar las primeras 5 iteraciones de cada función.

$$x^{(t+1)} = x^t - \alpha \nabla f(x^{(t)})$$

- Para la función f($\overrightarrow{x}$) $= 10(x_2 - x_1^2)^2 + (1 - x_1)^2$ tenemos las primeras 5 iteraciones:

  1) [0.02, 0.0]
  2) [3.9596*10e^{-2}, 8*10e^{-5}]
  3) [0.058781, 0.000377]
  4) [0.077533, 0.000993]
  5) [0.095827, 0.0019967]

  Donde el gradiente es el siguiente:

  $$\nabla f(x) = (-40x_2x_1 + 40x_1^3 + 2 + 2x_1, \quad 20x_2 - 20x_1^2)$$

Ya que el ejercicio pide las primeras 5 iteraciones las mostramos pero cabe recalcar que al aumentar las mismas el método no da un buen resultado debido a que no converge por lo que no ayudo mucho este método.

- Para la función f($\overrightarrow{x}$) $= 30 + \sum_{i=1}^{2}(x_i^2 - 10cos(2\pi x_i))$ tenemos las primeras 5 iteraciones:

  1) [0.0, 0.0]

  Donde el gradiente es el siguiente:

  $$\nabla f(x^{(t)}) = (2x_1 + 20\pi sen(2\pi x_1), \quad 2x_2 + 20\pi sen(2\pi x_2))$$

  Al realizar las iteraciones no llegamos a un buen resultado debido a nuestro punto de inicio, si prestamos atención al evaluar el gradiente en este punto nos resulta cero por lo que no vamos a avanzar del mismo punto inicial y el método se detiene.

- Para la función f($\overrightarrow{x}$) $= (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$ tenemos las primeras 5 iteraciones:

  1) [0.34, 0.38]
  2) [0.6156, 0.6948]
  3) [0.83845, 0.956072]
  4) [1.01812, 1.173388]
  5) [1.162441, 1.354599]

  Donde el gradiente es el siguiente:

  $$\nabla f(x) = (2(x_1 + 2x_2 - 7) + 4(2x_1 + x_2 - 5), \quad 4(x_1 + 2x_2 - 7) + 2(2x_1 + x_2 - 5))$$

  Ya que el ejercicio pide las primeras 5 iteraciones las mostramos pero se realizo la tarea de aumentar las iteraciones y se observa cierta convergencia a los valores [1, 3], ya que al solicitar 150 iteraciones nos dio como resultado [0.99949, 2.9984]

- Para la función f($\overrightarrow{x}$) $= 0.25(x_1^2 + x_2^2)$ - $0.5(x_1 x_2)$ tenemos las primeras 5 iteraciones:

  1) [0.0, 0.0]

  Donde el gradiente es el siguiente:

  $$\nabla f(x) = (0.\,25(2x_1) - 0.\,5x_2, \quad 0.\,25(2x_2) - 0.\,5x_1)$$

  Al realizar las iteraciones no llegamos a un buen resultado debido a nuestro punto de inicio, si prestamos atención al evaluar el gradiente en este punto nos resulta cero por lo que no vamos a avanzar del mismo punto inicial y el método se detiene.

Inciso b

Nos piden ocupar el método de Newton tenemos lo siguiente:

$$x^{(t+1)} = x^{(t)} - \frac{\nabla f(x^{(t)})}{H_f(x^{(t)})}$$

- Para la función f($\overrightarrow{x}$) $= 10(x_2 - x_1^2)^2 + (1 - x_1)^2$ tenemos las unicas iteraciones:

  1) [1.0, 0.0]

  2) [1.0, 1.0]

  Donde el gradiente y matriz Hessiana, son:

  $$\nabla f(x) = (-40x_2x_1 + 40x_1^3 + 2 + 2x_1, \quad 20x_2 - 20x_1^2)$$
  $$H = \begin{pmatrix} -40x_2 + 120x_1^2 + 2 & -40x_1 \\ -40x_1 & 20 \end{pmatrix}$$

  Es fácil observar que el método converge muy veloz por lo que solo se mostraron 2 iteraciones.

- Para la función f($\overrightarrow{x}$) = $30 + \sum_{i=1}^{2}(x_i^2 - 10cos(2\pi x_i))$ tenemos la siguiente iteración:

  1) [0.0, 0.0]

  Donde el gradiente y matriz Hessiana, son:

$$\nabla f(x^{(t)}) = (2x_1 + 20\pi sen(2\pi x_1), \quad 2x_2 + 20\pi sen(2\pi x_2))$$
$$H = \begin{pmatrix} 2 + 40\pi^2 cos(2\pi x_1) & 0 \\ 0 & 2 + 40\pi^2 cos(2\pi x_2) \end{pmatrix}$$

  Al realizar las iteraciones no llegamos a un buen resultado debido a nuestro punto de inicio, si prestamos atención al evaluar el gradiente en este punto nos resulta cero por lo que no vamos a avanzar del mismo punto inicial y el método se detiene, la matriz Hessiana no nos ayuda de mucho si el grandiente nos da cero.

- Para la función f($\overrightarrow{x}$) $= (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$ tenemos la unica iteración que se realizo:

  1) [1.0, 3.0]

  Donde el gradiente y matriz Hessiana, son:

$$\nabla f(x) = (2(x_1 + 2x_2 - 7) + 4(2x_1 + x_2 - 5), \quad 4(x_1 + 2x_2 - 7) + 2(2x_1 + x_2 - 5))$$
$$H = \begin{pmatrix} 10 & 8 \\ 8 & 10 \end{pmatrix}$$

  Se observa que el método es muy bueno sin importar el punto de inicio ya que nos converge al resultado en solo una iteración.

- Para la función f($\overrightarrow{x}$) $= 0.25(x_1^2 + x_2^2)$ - $0.5(x_1 x_2)$ tenemos las primeras 5 iteraciones:

  1) [0.0, 0.0]

Donde el gradiente y matriz Hessiana, son:

$$\nabla f(x) = (0.25(2x_1) - 0.5x_2, \quad 0.25(2x_2) - 0.5x_1)$$
$$H = \begin{pmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{pmatrix}$$

Al realizar las iteraciones no llegamos a un buen resultado debido a nuestro punto de inicio que en este caso al evaluarlo en el gradiente nos resulta cero y no nos permite avanzar.

Inciso c

Para este inciso solo nos pide aproximar el gradiente por lo que se va a omitir la expresión y solo se mostraran los resultados.

$$\frac{\partial f}{\partial x_i} = \frac{f(x + he_i) - f(x - he_i)}{2h}$$

- Para la función f($\overrightarrow{x}$) $= 10(x_2 - x_1^2)^2 + (1 - x_1)^2$ tenemos las primeras 5 iteraciones:
  1) [0.02, 0.0]
  2) [3.9496*$10e^{-2}$, 8*$10e^{-5}$]
  3) [0.058681, 0.000376]
  4) [0.077463, 0.000994]
  5) [0.095767, 0.0019867]

  Los resultados son similares a los obtenidos en el inciso a) debido a que solo estamos aproximando el gradiente y este mismo se esta ocupando para realizar las iteraciones.

- Para la función f($\overrightarrow{x}$) $= 30 + \sum_{i=1}^{2}(x_i^2 - 10cos(2\pi x_i))$ tenemos las primeras 5 iteraciones:

1) [0.0, 0.0]

Nos volvio a suceder lo mismo que en el inciso a), nos quedamos en el mismo punto debido al punto de inicio.

- Para la función f($\overrightarrow{x}$) = $(x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$ tenemos las primeras 5 iteraciones:

  1) [0.34, 0.38]
  2) [0.6156, 0.6948]
  3) [0.83845, 0.956072]
  4) [1.01812, 1.173388]
  5) [1.162441, 1.354599]

  Vuelve a darnos los mismos valores por lo que para en este caso no cambia mucho el ocupar la aproximación o el gradiente que previamente calculamos.

- Para la función f($\overrightarrow{x}$) $= 0.25(x_1^2 + x_2^2)$ - $0.5(x_1 x_2)$ tenemos las primeras 5 iteraciones:

  1) [0.0, 0.0]

  El resultado sigue siendo el mismo por lo que ya mencionamos antes, el punto de inicio no favorece mucho este otro método.

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

b) f($\overrightarrow{x}$) = $(1-x_1)^2$ $+100(x_2-x_1^2)^2$, sujero a $x_1^2+x_2^2-2 \leq 0$.

Solución

Inciso a

Vamos a encontrar las derivadas parciales de $f(x), g(x)$ y $h(x)$, así tenemos: (h(x) y g(x) corresponde a las restricciones)

$$\nabla f(x) = \begin{pmatrix} -2(1-x_1)-400x_1(x_2-x_1^2) \\ 200(x_2-x_1^2) \end{pmatrix}$$

$$\nabla h(x) = \begin{pmatrix} 3(x_1-1)^2 \\ -1 \end{pmatrix}$$

$$\nabla g(x) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Al tener las derivadas parciales correspondiente nuestra tarea que queremos realizar es encontrar los valores lambda que nos ayuden a cumplir:

$$min \;\; (1-x_1)^2+100(x_2-x_1^2)^2+\lambda_1((x_1-1)^3-x_2)+\lambda_2(x_1+x_2-2)$$

Para solucionar nuestro problema vamos a realizar casos: a) $\lambda_1=\lambda_2=0$, b) $\lambda_1=0;\; \lambda_2\neq 0$, $\lambda_1\neq 0;\; \lambda_2=0$ y $\lambda_1\neq 0;\; \lambda_2\neq 0$.

Para nuestro primer caso a) como los valores de lambda son cero lo unico que nos queda es $\nabla f(x)=0$, al desarollar tenemos:

$$-2(1-x_1)-400x_1(x_2-x_1^2)=0 \;\; \ldots(1)$$
$$200(x_2-x_1^2)=0 \;\; \ldots(2)$$

De (2) podemos ver que $x_2=x_1^2$ por lo que al sustituirlo en 1 tenemos que $x_1=1$ y con este valor encontramos $x_2=1$, una vez que tenemos nuestra

solución para este primer caso debemos observar si cumple las condiciones necesarias para ser el optimo.
Las condiciones son las siguientes:
Condición 1:

$$(x_1 - 1)^3 - x_2 \leq 0$$
$$(1 - 1)^3 - 1 \leq 0; \quad \textit{si cumple}$$

Condición 2:

$$x_1 + x_2 - 2 \leq 0$$
$$1 + 1 - 2 \leq 0; \quad \textit{si cumple}$$

Observamos que cumple nuestras condiciones necesarias para poder ser optimo. Al tomar los valores de lambda iguales a cero nos dice que nuestras condiciones están inactivas, es decir, que alcanzamos nuestro valor optimo sin necesidad de contemplarlas ya que nuestra solución es un optimo y sigue cumpliendo estas condiciones por lo que no es necesario resolver los otros casos.

Inciso b

Vamos a encontrar las derivadas parciales de $f(x)$ y $g(x)$, así tenemos:
(g(x) corresponde a las restricciones)

$$\nabla f(x) = \begin{pmatrix} -2(1 - x_1) - 400x_1(x_2 - x_1^2) \\ 200(x_2 - x_1^2) \end{pmatrix}$$

$$\nabla g(x) = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix}$$

Al tener las derivadas parciales correspondiente nuestra tarea que queremos realizar es encontrar los valores lambda que nos ayuden a cumplir:

$$min \;\; (1 - x_1)^2 + 100(x_2 - x_1^2)^2 + \lambda_1(x_1^2 + x_2^2 - 2)$$

Para este inciso tenemos dos casos ya que solo contamos con un lambda, los siguientes: $\lambda_1 = 0$ y $\lambda_1 \neq 0$.
Para el primer caso:

$$-2(1-x_1) - 400x_1(x_2 - x_1^2) = 0 \quad \ldots(1)$$
$$200(x_2 - x_1^2) = 0 \quad \ldots(2)$$

Volvemos a tener la solución que el inciso a) por lo que solo nos resta observar la condición. La solución es: $x_1 = x_2 = 1$ Condición 1:

$$x_1^2 + x_2^2 - 2 \leq 0$$
$$1 + 1 - 2 \leq 0; \quad \textit{si cumple}$$

Nuevamente observamos que nuestra solución $x_1 =$ y $x_2 = 1$ es un optimo por lo que resolver el segundo caso($\lambda_1 > 0$) no sera necesario ya que cumple las condiciones y es optimo.

3. Problema 3

Dado la descomposición del problema de optimización de la SVM:

$$min \quad \frac{1}{2}\alpha_B^t Q_{BB}\alpha_B - (e_B^t - \alpha_N Q_{NB})\alpha_B + (\frac{1}{2}\alpha_N^t Q_{NN}\alpha_N - e_N^t \alpha_N)$$

$$subect \quad to: \alpha_B^t y_B = z$$

donde $z = -\alpha_N^t y_N$ 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 α_i de la Ecuación (1) es equivalente a:

$$\nabla\hat{\alpha}_i = \begin{cases} \sum_{k=1}^{n}(\alpha_i\alpha_k Q_{i,k} + \alpha_i\alpha_k Q_{jk}) - 2; & si\ y_i = y_j \\ \sum_{k=1}^{n}(\alpha_i\alpha_k Q_{i,k} - \alpha_i\alpha_k Q_{jk}); & si\ y_i \neq y_j \end{cases}$$

Solución

Para poder desarrollar lo que nos pide el ejercicio vamos a mostrar como vamos a colocar nuestra matriz Q, donde vamos a dividirla en cuatro bloques, en los bloques vamos a incluir las variables basicas y no basicas.

$$Q = \begin{pmatrix} Q_{ii} & Q_{ij} & | & Q_{i1} & Q_{i2} & ... & Q_{in} \\ Q_{ji} & Q_{jj} & | & Q_{j1} & Q_{j2} & ... & Q_{jn} \\ -- & -- & -- & -- & -- & -- & -- \\ Q_{1i} & Q_{1j} & | & Q_{11} & Q_{12} & ... & Q_{1,n} \\ Q_{2i} & Q_{2j} & | & Q_{21} & Q_{22} & ... & Q_{2,n} \\ \vdots & \vdots & | & \ddots & \ddots & ... & \vdots \\ Q_{ni} & Q_{nj} & | & Q_{n1} & Q_{n2} & ... & Q_{nn} \end{pmatrix}$$

Realizamos lo mismo con nuestra matriz que tiene las variables.

$$\alpha = \begin{pmatrix} \alpha_i \\ \alpha_j \\ ---- \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}$$

Si observamos solo tomamos los renglones i y j para los colocamos al principio y respetar los demás. Con esta nueva forma de las matrices Q y α, vamos a poder representar la siguiente:

$$Q = \begin{pmatrix} Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN} \end{pmatrix}; \quad \alpha = \begin{pmatrix} \alpha_B \\ \alpha_N \end{pmatrix}$$

Ya que para poder solucionar el problema nos vamos a basar en lo siguiente:

$$\frac{1}{2}\alpha_B^t Q_{BB}\alpha_B - (e_B^t - \alpha_N Q_{NB})\alpha_B + (\frac{1}{2}\alpha_N^t Q_{NN}\alpha_N - e_N^t\alpha_N)$$
$$\frac{1}{2}\alpha_B^t Q_{BB}\alpha_B - (e_B^t - \alpha_N Q_{NB})\alpha_B + constante$$

Donde el último término lo tomamos como una constante por lo que realmente solo trabajariamos con los dos primeros, en este caso tendremos:

$$\frac{1}{2}\alpha_B^t Q_{BB}\alpha_B - (e_B^t - \alpha_N Q_{NB})\alpha_B \quad ...(1)$$

Vamos a analizar el producto que nos presentan en $\alpha_N^t Q_{NB}$ antes de pasar a sustituir y resolver.

$$\alpha_N^t Q_{NB} = \begin{pmatrix}\alpha_1 & \alpha_2 & \alpha_3 & ... & \alpha_n\end{pmatrix} \begin{pmatrix} Q_{1i} & Q_{1j} \\ Q_{2i} & Q_{2j} \\ \vdots & \vdots \\ Q_{ni} & Q_{nj} \end{pmatrix}$$

Al realizar dicho producto vamos a tener:

$$\alpha_N^t Q_{NB} = \begin{pmatrix}\alpha_1(Q_{1i}) + \alpha_2(Q_2 i) + ... + \alpha_n(Q_{ni}) & \alpha_1(Q_{1j}) + \alpha_2(Q_2 j) + ... + \alpha_n(Q_{nj})\end{pmatrix}$$

Donde la matriz que nos resulto fue un matriz de 1X2, ahora vamos a sustituir en nuestra ecuación (1).

$$\frac{1}{2}\alpha_B^t Q_{BB}\alpha_B - (e_B^t - \alpha_N Q_{NB})\alpha_B$$

$$\frac{1}{2}\begin{pmatrix}\alpha_i & \alpha_j\end{pmatrix}\begin{pmatrix} Q_{ii} & Q_{ij} \\ Q_{ji} & Q_{jj}\end{pmatrix}\begin{pmatrix}\alpha_i \\ \alpha_j\end{pmatrix} - \left((1 \;\; 1) - \alpha_N Q_{NB}\right)\begin{pmatrix}\alpha_i \\ \alpha_j\end{pmatrix}$$

$$
\begin{aligned}
\frac{1}{2}[\ Q_{ii}\alpha_i^2 + 2Q_{ij}\alpha_i\alpha_j + Q_{jj}\alpha_j^2\ ] - [\ (\alpha_i + \alpha_j) - ... \\
... - \alpha_i[\alpha_1(Q_{1i}) + \alpha_2(Q_2 i) + ... + \alpha_n(Q_{ni})] - ... \\
... - \alpha_j[\alpha_1(Q_{1j}) + \alpha_2(Q_2 j) + ... + \alpha_n(Q_{nj})]\ ]
\end{aligned}
$$

simplificando un poco, tenemos:

$$
\begin{aligned}
\frac{1}{2}[\ Q_{ii}\alpha_i^2 + 2Q_{ij}\alpha_i\alpha_j + Q_{jj}\alpha_j^2\ ] - [\ (\alpha_i + \alpha_j) - ... \\
... - \alpha_i[\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{ki})] - \alpha_j[\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{kj})]\ ]
\end{aligned}
$$

Nota: En ambas sumatorias no pasan por i y j.
Una vez que tenemos lo anterior vamos a revisar dos casos, uno donde $K = \alpha_i + \alpha_j$ donde sabemos que $y_i = y_j$ y $K = \alpha_i - \alpha_j$ donde sabemos que $y_i \neq y_j$.
Para el primer caso tenemos que $\alpha_j = K - \alpha_i$ donde sabemos que $y_i = y_j$, sustituimos y resolvemos.

$$
\begin{aligned}
\frac{1}{2}[\ Q_{ii}\alpha_i^2 + 2Q_{ij}\alpha_i(K - \alpha_i) + Q_{jj}(K - \alpha_i)^2\ ] - [\ (\alpha_i + (K - \alpha_i)) - ... \\
... - \alpha_i[\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{ki})] - (K - \alpha_i)[\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{kj})\ ]
\end{aligned}
$$

Una vez que tenemos esto vamos a encontrar la derivada parcial respecto de α_i, así:

$$
\begin{aligned}
\nabla_{\alpha_i} = Q_{ii}\alpha_i - 2Q_{ij}\alpha_i + Q_{ij}K + Q_{jj}K - Q_{jj}\alpha_i \\
+[\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{ki})] - [\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{kj})\ ]
\end{aligned}
$$

Si observamos podemos completar terminos sustituyendo el valor de K que teniamos en un principio, $K = \alpha_i + \alpha_j$, por lo que vamos a tener:

$$
\nabla_{\alpha_i} = \sum_{k=1}^{n}(\alpha_k Q_{ki} - \alpha_k Q_{kj})
$$

Observe que ya se logro completar ambas sumas que en un principio no pasaban por $k = i$ y $k = j$.
Pero como la matriz Q es simetrica podemos hacer lo siguiente:

$$\nabla_{\alpha_i} = \sum_{k=1}^{n} (\alpha_k Q_{ik} - \alpha_k Q_{jk})$$

Para el segundo caso tenemos que $\alpha_j = \alpha_i - K$ donde sabemos que $y_i \neq y_j$, sustituimos y resolvemos.

$$\frac{1}{2}[\ Q_{ii}\alpha_i^2 + 2Q_{ij}\alpha_i(\alpha_i - K) + Q_{jj}(\alpha_i - K)^2\ ] - [\ (\alpha_i + (\alpha_i - K)) - ...$$
$$... - \alpha_i[\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{ki})] - (\alpha_i - K)[\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{kj})\ ]$$

Una vez que tenemos esto vamos a encontrar la derivada parcial respecto de α_i, así:

$$\nabla_{\alpha_i} = Q_{ii}\alpha_i + 2Q_{ij}\alpha_i - Q_{ij}K - Q_{jj}K + Q_{jj}\alpha_i - 2$$
$$+[\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{ki})] + [\sum_{k=1,k\neq i,k\neq j}^{n} \alpha_k(Q_{kj})\ ]$$

Si observamos podemos completar terminos con el valor de $K = \alpha_i - \alpha_j$, por lo que vamos a tener:

$$\nabla_{\alpha_i} = \sum_{k=1}^{n} (\alpha_k Q_{ki} + \alpha_k Q_{kj}) - 2$$

Al igual que en el primer caso acabamos de completar los terminos de las sumas.

Pero como la matriz Q es simetrica podemos hacer lo siguiente:

$$\nabla_{\alpha_i} = \sum_{k=1}^{n} (\alpha_k Q_{ik} + \alpha_k Q_{jk}) - 2$$

al observar nuestros resultamos llegamos a la conclusión que lo que propone como gradiente el problema no es correcto y los resultados correctos se mostraron.

$$\nabla \hat{\alpha}_i = \begin{cases} \sum_{k=1}^{n} (\alpha_k Q_{i,k} + \alpha_k Q_{jk}) - 2; & si\ y_i \neq y_j \\ \sum_{k=1}^{n} (\alpha_i \alpha_k Q_{i,k} - \alpha_i \alpha_k Q_{jk}); & si\ y_i = y_j \end{cases}$$

4. Problema 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 coecientes.

Solución

Nuestro problema lo vamos a plantear de la siguiente forma:

$$\sum_{i=1}^{m} (w^t x_i - Y_i)^2 \rightarrow ||w^t x_i - Y_i||^2$$

Donde w es la matriz que tiene los coeficientes que nos ayudan a realizar la aproximación y Y_i es el vector resultado que ya conocemos.
Se desea minimizar $||w^t x_i - Y_i||^2$ ya que queremos tener una buena aproximación por parte de la matriz w pero agregamos una penalización a los

coeficientes de la matriz w.
Al final tenemos como problema a optimizar:

$$\begin{aligned} min \;\; & ||w^t x_i - Y_i||^2 \\ s.a. \;\; & ||w|| < t \end{aligned}$$

$$||w^t x_i - Y_i||^2 - \lambda ||w||$$

El programa que se implemento fue con ayuda de gradiente en desceso simultaneo, la implementación se anexa y se explica poco a poco el proceso.