Tarea 5. Optimización

Oscar Esaú Peralta Rosales
Maestría en Computación
Centro de Investigación en Matemáticas

***Resumen*—A continuación se presentan la implementación y resultados de dos algoritmos de región de confianza, el primero el Algoritmo de *Dogleg* y el segundo una modificación al algoritmo de obtención del paso y derección de descenso a través de usar descenso de gradiente sobre el modelo m_k a optimizar. Por último se presenta la solución a un problema de segmentación a través del histograma generado por los colores de una imagen y se compara contra una aproximación a dicho histograma usando funciones de base radial (en este caso la gaussiana) a través de la optimización de las medias y los coeficientes de ponderación de cada función base radial.**

***Index Terms*—Región de Confianza, Dogleg, Funciones de Base Radial**

I. Introduction

Al igual que los métodos de búsqueda en linea los métodos basados en región de confianza generan cada paso a través de un modelo cuadrático de la función objetivo, en dónde se define un región alrededor del valor de la iteración actual y se elige un paso (junto con su tamaño) que minimize el modelo en la en dicha región bajo la premisa de que el modelo es una buena representación de la función objetivo en la región. El tamaño de la región se actualiza de acuerdo a el rendimiento de la iteración anterior, si el modelo es bastante confiable y produce pasos consistentes en el la región puede ser incrementada de tamaño. Si no se encuentra un paso válida entonces el tamaño de la región se decremente hasta encontrar alguno [1].

El modelo está dado por:

$$m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p \tag{1}$$

Dónde B_k es el Hessiano de f. Para obtener cada paso entonces se busca minimizar el problema

$$\min_{p \in \mathcal{R}^n} m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p, \ ||p|| \leq \Delta_k \tag{2}$$

Donde Δ_k es el radio de la región de confianza. Este radio es incrementado y reducido dependiendo del cociente

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} \tag{3}$$

Si $\rho_k < \frac{1}{4}$ decrementamos el radio a un cuarto de su valor, si $\rho_k > \frac{3}{4}$ elegimos $\Delta_{k+1} = min(2\Delta_k, \hat{\Delta})$, con $\hat{\Delta}$ como el radio máximo.

Uno de los métodos para la obtención del tamaño de paso y la dirección de descenso p_k es a través del método de *Dogleg*. Se elige remplazar la curva de trayectoria óptima de la solucion que minimiza a m_k, por dos segmentos desde el origen del minimizador a el punto p_k^u y del punto $p_k^U = -\frac{g_k^t g_k}{g_k^t B_k g_k}$ al punto p_k^B, dónde $p_k^B = -B_k^{-1} g_k$ es la solucion óptima al problema sin restricciones (2).

Si $\Delta_k >= ||p_k^B||$ la solucion es precisamente $p_k = p_k^B$, si $\Delta_k >= ||p_k^U||$ la solución es $p_k = p_k^U \frac{\Delta_k}{||p_k^U||}$, de lo contrario la solución es la proporcion τ de p_k^B de la intersección de la región de confianza con el segmento p_k^U a p_k^B.

La segunda forma de obtención del paso y dirección de descenso p_k se describe el algoritmo 1 del Apéndice. La idea central es aplicar descenso de gradiente sobre m_k para obtener un nuevo punto z_j que minimize m_k a través de la direción d_j, en el momento que $||z_j|| \geq \Delta_k$ o $d_j^T B_k d_j \leq 0$ buscamos la intersección de la región de confianza con el segmento z_j a d_j tal que $||z_j + \tau d_j|| = \Delta_k$ con $p_k = z_j + \tau d_j$.

La función a optimizar para aproximar un histograma en 3D a través de funciones base con kernel gaussiano está dada por minimizar siguiente error:

$$\min_{\alpha^j, \mu^j} g(\alpha^j, \mu^j)$$

$$g(\alpha^j, \mu^j) = \sum_{c \in \Omega} \Big[h^j(c) - \sum_{i}^{n} \alpha_i^j exp\big(\frac{-||c - \mu_i^j||_2^2}{2\sigma^2}\big) \Big]^2 \tag{4}$$

Donde:

- $j \in \{1, 2\}$ corresponden al fondo y al objeto.
- $c = [r, g, b]^T$, con $r \in B_1 = \{1, \ldots, b_1\}$, $g \in B_2 = \{1, \ldots, b_2\}$ y $b \in B_2 = \{1, \ldots, b_2\}$ y b_1, b_2, b_3 son el número de bins en cada cana RGB.
- $\Omega = B_1 x B_2 X B_3$, donde $XxY := \{(x.y) : x \in X, y \in Y\}$ representa el producto cartesiano.
- $h^j(r, g, b)$ es el histograma en 3D de la clase j.
- n es el número de elementos de la base radial.
- $\alpha^j = [\alpha_1^j], \ldots, \alpha_1^n$ son los pesos de la combinación lineal de la base radial
- $mu^j = [mu_1^j, \ldots, mu_n^j]$ son las medias de cada funcion base.
- σ es un parámetro ajustable.

Asi, buscamos optimizar los valores de α^j y μ^j tal que el error sea mínimo.

II. Métodología

II-A. Algoritmo de Región de Confianza: Dogleg

Como se mencionó anteriormente los algoritmos de región de confianza se pueden dividir generalmente en dos pasos, el primero es buscar una dirección y tamaño de paso p_k que minimize el modelo $m_k(p_k)$ y luego encontrado p_k necesitamos decidir si podemos aumentar o disminuir la región de confianza dado el coeficiente ρ_k (Apéndice $V-A$-Algoritmo 1).

El algoritmo de Dogleg se describe en el Apendice $V-A$-Algoritmo 2. Se probó usando la función de Rosembrock con n = 100 y la función de Wood.

II-B. Algoritmo de Región de Confianza: Descenso de gradiente sobre el modelo

Para realizar la implementación de este variante se procedió primero a realizar el calculo del valor de τ tal que $||z_j + \tau d_j|| = \Delta_k$, expandiendo el tenemos que $d_j^T d_j \tau^2 + 2z_j^T d_j \tau + z_j^T z_j - \Delta_k^2 = 0$, hacemos $a = d_j^T d_j$, $b = 2z_j^T d_j$ y $c = z_j^T z_j - \Delta_k^2$, así podemos obtenemos lo coeficientes de una ecuación cuadrática y resolver para τ usando la fórmula general.

$\alpha_j = arg\,\text{mín}_{\alpha>0}\, m_k(z_j + \alpha_j d_j)$, derivando e igualando a cero tenemos que $\alpha_j = -\frac{g_k^T d_j + z_j^T B_k d_j}{d_j^T B_k d_j}$. El algoritmo completo implementado se encuentra en el Apéndice $V-A$.

II-C. Aproximación de histograma en 3D a través de funciones base radial con kernel gaussiano

Los parámetros α^j y μ^j a optimizar de la función (4) son obtenidos optimizando por separado fijando α^j y luego μ^j hasta convergencia:

$$\alpha_{k+1}^j = arg \min_{\alpha^j \in (R)^n} g(\alpha^j, \mu_k^j)$$

$$\mu_{k+1}^j = arg \min_{\mu^j \in (R)^n} g(\alpha_k^j, \mu^j)$$

Notemos que para realizar la optimización necesitamos calcular dos gradientes un con respecto a α^j y otro con respecto a μ^j donde cada entrada del gradiente está dadas por:

$$\frac{\partial}{\partial_{\alpha_k^j}} - 2\sum_{c\in\Omega}\left[h^j(c) - \sum_i^n \alpha_i^j exp^{-w_i}\right]\exp^{w_k}$$

y

$$\frac{\partial}{\partial_{\mu_k^j}} - \frac{2}{\sigma^2}\sum_{c\in\Omega}\left[h^j(c) - \sum_i^n \alpha_i^j \exp^{-w_i}\right]\alpha_k \exp^{-w_k}(c - \mu_k^j)$$

donde $w_i = \frac{||c-\mu_i^j||_2^2}{2\sigma^2}$

Durante la implementación se utilizó el algoritmo de optimización de Barzilai-Borwein, nótese que para otros algoritmos es necesario calcular el Hessiano.

La generación del histograma 3D de una imágen se genera con ayuda de un script en python el cuál nos permite marcar con dos colores rojo y azul, el objeto y el fondo. Posteriormente para cada pixel de alguna imagen se mapea a su correspondiente valor c y después de la optimización sobre el fondo y el objeto se elige el color mediante:

$$f(c, \alpha^j, \mu^j) = \sum_i^n \alpha_i^j exp\left(\frac{-||c - \mu_i^j||_2^2}{2\sigma^2}\right)$$

$$F(c, \alpha^1, \mu^1) = \frac{f(c, \alpha^1, \mu^1) + \epsilon}{f(c, \alpha^1, \mu^1) + f(c, \alpha^2, \mu^2) + 2\epsilon}$$

$$F(c, \alpha^2, \mu^2) = \frac{f(c, \alpha^2, \mu^2) + \epsilon}{f(c, \alpha^1, \mu^1) + f(c, \alpha^2, \mu^2) + 2\epsilon}$$

Con $\epsilon = 0{,}01$ y si $F(c, \alpha^1, \mu^1) < F(c, \alpha^2, \mu^2)$ asignar al pixel el color azul de lo contrario el color rojo.

Realizando lo mismo a través del histograma en 3D tenemos que:

$$H1(c) = \frac{h^1(c) + \epsilon}{h^1(c) + h^2(c) + 2\epsilon}$$

$$H2(c) = \frac{h^2(c) + \epsilon}{h^1(c) + h^2(c) + 2\epsilon}$$

Si $H1(c) < H2(c)$ asignar al pixel el color azul de lo contrario el color rojo.

III. Resultados

III-A. Algoritmo de Región de Confianza: Dogleg

En la tabla I se muestra una comparativa del rendimiento del método con las dos funciones. Nótese que los resultados obtenidos son para el mínimo local ya conocido.

Cuadro I
Tabla comparativa del método de Dogleg

Función	***Iteraciones***	$\|\|g_k\|\|$	$\|f(x_k) - f(x^*)\|$
Rosembrock	61	9.9986	3.9866
Wood	44	1.9180	7.8769

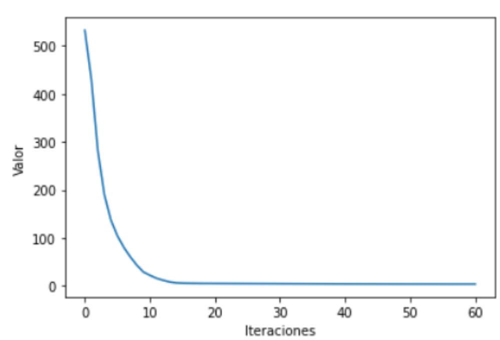

Figura 1. Decremento de la función objetivo a través de las iteraciones, Rosembrock

III-B. *Algoritmo de Región de Confianza: Descenso de gradiente sobre el modelo*

En la tabla II se muestra una comparativa del rendimiento del método con las dos funciones. Nótese que los resultados obtenidos son para el mínimo local ya conocido para el caso de la función de Rosembrock y al óptima para la función de Wood.

Cuadro II
TABLA COMPARATIVA DEL MÉTODO USANDO DESCENSO DE GRADIENTE SOBRE EL MODELO

Función	***Iteraciones***	$\|\|g_k\|\|$	$\|f(x_k) - f(x^*)\|$
Rosembrock	352	9.9986	3.9866
Wood	250	2.0000	4.2463e-13

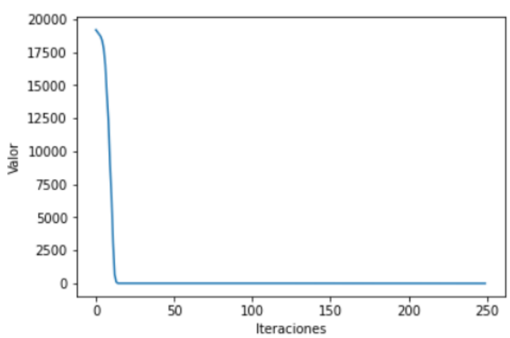

Figura 2. Decremento de la función objetivo a través de las iteraciones, Wood

III-C. *Aproximación de histograma en 3D a través de funciones base radial con kernel gaussiano*

Se realizaron las pruebas sobre diversas grupos de imágenes, todos los histogramas se obtuvieron a través del script en python brindado. Las siguiente imágenes muestran los mejores resultados obtenidos para las imágenes.

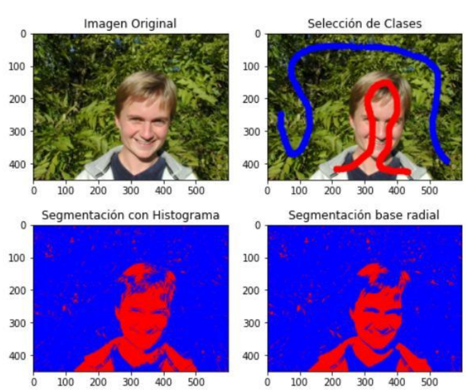

Figura 3. Segmentación a través del histograma y la aproximación a este, $bins = 3$, funciones base = 30, $\sigma = 20$

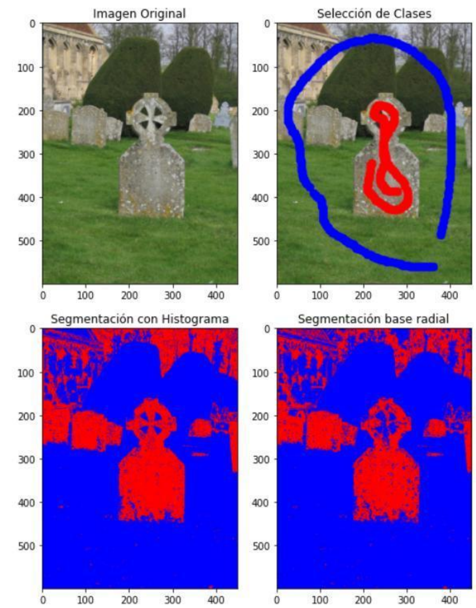

Figura 4. Segmentación a través del histograma y la aproximación a este, $bins = 3$, funciones base = 30, $\sigma = 0{,}1$

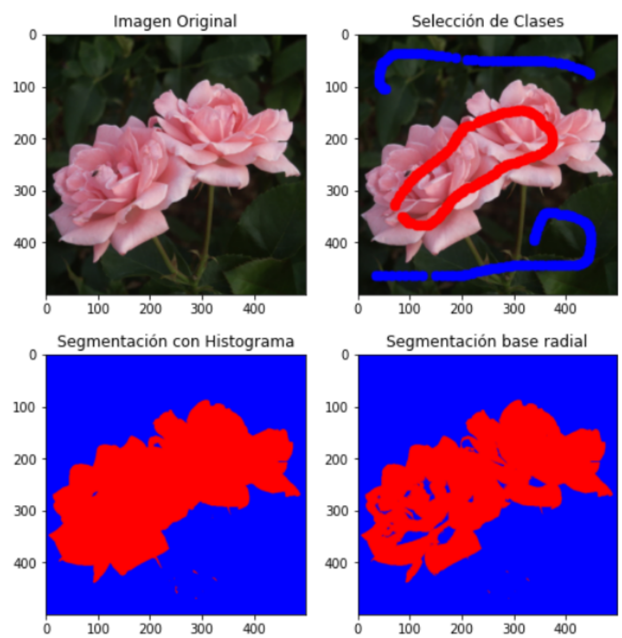

Figura 5. Segmentación a través del histograma y la aproximación a este, $bins = 3$, funciones base = 30, $\sigma = 0{,}1$

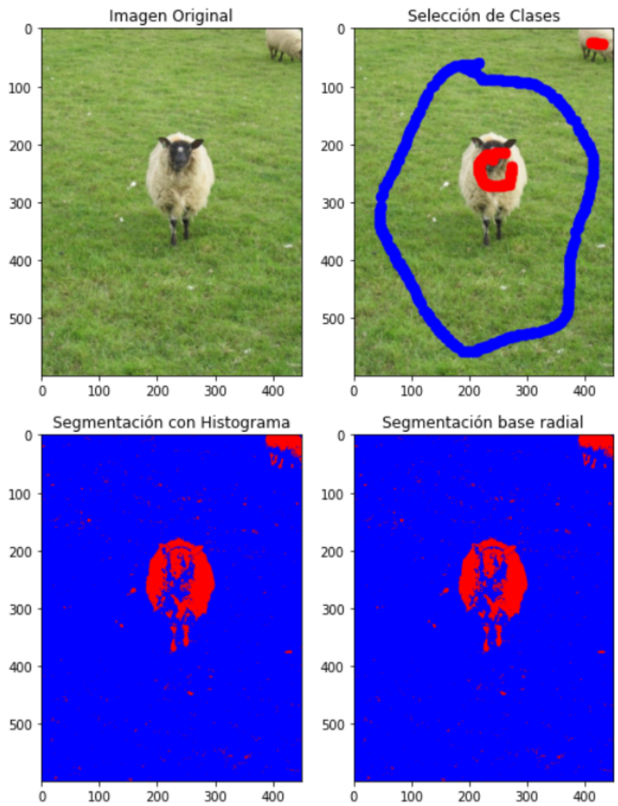

Figura 6. Segmentación a través del histograma y la aproximación a este, $bins = 3$, funciones base = 30, $\sigma = 0{,}1$

IV. Conclusiones

Como se mencionó anteriormente en los métodos de región de confianza el cálculo del tamaño de paso y la dirección de gradiente puede ser en diversas formas. En este caso se implementaron dos formas la primera a través del método de Dogleg, el cual tratar de aproximar la trayectoria de la solución mediante dos segmentos formados por el origen y el punto p_k^U y de p_k^U a el punto p_k^B. La segunda forma fue interar sobre el modelo hasta encontrado un tamaño de paso y dirección valido a través de descenso de gradiente sobre el modelo. Como se observa en los resultados la convergencia del método de Dogleg es mucho más rápida que la segunda aproximación covergiendo al rededor de 60 iteraciones vs 300 iteraciones del método usando descenso. En este caso con ambos métodos no se pudo obtener el mínimo óptimo llegando solo al local para la función de Rosembrock, y llegando al mínimo óptimo para la función de Wood solo en el segundo método de para la obtención de p_k.

Para varias imágenes, las aproximaciones a los histogramas fueron similares, mostrando imágenes segmentadas con muy pocas diferencias. Sin embargo, para poder alcanzar resultados buenos la clave es encontrar valores de σ que nos permitan aproximarla de la mejor manera. Por lo regular con alrededor de 60 a 80 funciones base los resultados son bastante buenos. El número de bins usandos para la generación de los histogramas es de 3, un número mayor de bins provoca un incremento substancial en el tiempo de ejecución y convergencia (horas).

V. Apéndice

V-A. Algoritmos

Algorithm 1: Algoritmo de Región de Confianza

Result: x_k
x_k <- Inicializar
while $||g_k|| > tol$ **do**
  Calcular p_k que minimize el modelo $m_k(p_k)$
  $\rho_k = \frac{f(x_k)-f(x_k+p_k)}{m_k(0)-m_k(p_k)}$
  if $\rho < \frac{1}{4}$ **then**
    $delta_k = \frac{1}{4}\Delta_k$
  end
  else if $\rho > \frac{3}{4}$ **then**
    $delta_k = min(2\Delta_k, \hat{\Delta})$
  end
  if $\rho \leq \eta$ **then**
    continue
  end
  $x_k = x_k + p_k$
end

Algorithm 2: Método de Dogleg, cálculo de p_k

Result: p_k
g_k <- Ingresar
B_k <- Ingresar
Δ_k <- Ingresar
while $||g_k|| > tol$ **do**
  $p_k^U = \frac{g_k^T g_k}{g_k^T B_k g_k}$
  $p_k^B = -B_k^{-1} g_k$
  if $||p_k^B \leq \Delta_k||$ **then**
    $p_k = p_k^B$
  end
  if $||p_k^U \geq \Delta_k||$ **then**
    $p_k = p_k^U \frac{\Delta_k}{||p_k^U||}$
  end
  else
    $a = ||p_k^B - p_k^U||^2$
    $b = 2p_k^B(p_k^B - p_k^U)$
    $c = ||p_k^U||^2 - \Delta^2$
    $\lambda = \frac{-b+-\sqrt{b^2-4ac}}{2a} : \lambda \in [0,1]$
    $\tau = \lambda + 1$
    $p_k = p_k^U + (\tau - 1{,}0) * (p_k^B - p_k^U)$
  end
end

Algorithm 3: Descenso de gradiente sobre $m_k(p)$ para cálculo de p_k

Result: p_k
$j = 0$
$z_0 = 0$
$d_0 = -\nabla m_k(z_0)$
g_k <- Ingresar
while $||d_j|| \neq 0, j < J$ **do**
    if $||d_j^T B_k d_j \leq 0||$ **then**
        $a = d_j^T d_j$
        $b = 2z_j^T d_j$
        $c = z_j^T z_j - \Delta^2$
        $\tau = \frac{-b+-\sqrt{b^2-4ac}}{2a} : \tau \geq 0$
        return $p_k = z_j + \tau d_j$
    end
    $\alpha_j = -\frac{g_k^T d_j + z_j^T B_k d_j}{d_j^T B_k d_j}$
    $z_{j+1} = z_j + \alpha_j d_j$
    if $||z_{j+1}|| > \Delta_k$ **then**
        $a = d_j^T d_j$
        $b = 2z_j^T d_j$
        $c = z_j^T z_j - \Delta^2$
        $\tau = \frac{-b+-\sqrt{b^2-4ac}}{2a} : \tau \geq 0$
        return $p_k = z_j + \tau d_j$
    end
    $d_j = -\nabla m_k(z_{j+1})$
    $j+ = 1$
end
$p_k = z_{j+1}$ return p_k

Referencias

[1] Jorge Nocedal, Stephen J. Wright, "Numerical Optimization," Second Edition, Springer.