OPTIMIZATION. HOMEWORK 2

OSCAR DALMAU

1. El conjunto $S = \{a \in \mathbb{R}^k |\ p(0) = 1,\ |p(t)| \leq 1 \text{ for } t \in [\alpha, \beta]\}$, donde $a = [a_1, \cdots, a_k]^T$ y
$$p(t) \quad = \quad a_1 + a_2 t + \cdots + a_k t^{k-1},$$
es convexo?. Argumente su respuesta
2. Suponga que f es convexa, $\lambda_1 > 0$ y $\lambda_2 \leq 0$ con $\lambda_1 + \lambda_2 = 1$, y sean $x_1,\ x_2 \in \text{dom } f$. Muestra que la desiguialdad
$$f(\lambda_1 x_1 + \lambda_2 x_2) \quad \geq \lambda_1 f(x_1) + \lambda_2 f(x_2)$$
siempre se cumple.
3. Muestra que la función $f : \mathbb{R}^n \to \mathbb{R}$
$$f(x) \quad = \quad -\exp(-g(x))$$
es convexa, donde $g : \mathbb{R}^n \to \mathbb{R}$ tiene un dominio convexo y cumple
$$\begin{bmatrix} \nabla^2 g(x) & \nabla g(x) \\ \nabla^T g(x) & 1 \end{bmatrix} \quad \succeq \quad 0$$
para $x \in \text{dom } g$.
4. Implementar el método de gradiente descendente (usando la dirección de máximo descenso) en uno de los siguientes lenguajes Python/C/C++ (preferiblemente en Python). La implementación permitirá usar diferentes los siguientes tamaños de paso vistos en clase:
   - Tamaño fijo $\alpha_k = \alpha$,
   - Aproximacion $\alpha_k = \frac{g_k^T g_k}{g_k^T H_k g_k}$
   - Usando Backtracking

   que podrán ser proporcionados a la función a través de un parámetro o parámetros.

Para los experimentos considerar las siguientes **especificaciones generales**. En total deberán realizar y mostrar los resultados de experimentos para cada función.

- resolver usando dos puntos iniciales x_0 diferentes, uno de ellos sera el punto inicial dado en cada problema y el otro lo pueden escoger aleatoriamente y lejano al óptimo.

CUADRO 1. Ejemplo de tabla. N denota el número de iteraciones que realizó el algoritmo, y f representa la función objetivo del problema.

k	$\|\|x_{k+1} - x_k\|\|$	$\|\|\nabla f(x_k)\|\|$	$f(x_k)$
1	val	val	val
2	val	val	val
3	val	val	val
N-2	val	val	val
N-1	val	val	val
N	val	val	val

- reportar cada experimento en el informe escrito, y seguir un formato de tabla similar a el cuadro 1.
- agregar en el informe escrito, las conclusiones para cada experimento y comentar que observan según sus resultados obtenidos.

Conjunto de Funciones

Problema 1 (Rosembrock, $n = 2$): Extended Rosembrock function, for $n = 2$

$$\begin{aligned} f(\boldsymbol{x}) &= \left[100(x_2 - x_1^2)^2 + (1 - x_1)^2\right] \\ \boldsymbol{x}^0 &= [-1.2, 1]^T \\ \boldsymbol{x}^* &= [1, 1]^T \\ f(\boldsymbol{x}^*) &= 0 \end{aligned}$$

Para este problema, tener en cuenta que:

- Deberán visualizar la gráfica de las curvas de nivel junto con los puntos generados por el algoritmo y la curva o camino recorrido por el método

Problema 2 (Rosembrock, n = 100): Extended Rosembrock function, for $n = 100$

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right] \\ \boldsymbol{x}^0 &= [-1.2, 1, 1, \cdots, 1, -1.2, 1]^T \\ \boldsymbol{x}^* &= [1, 1, \cdots, 1, 1]^T \\ f(\boldsymbol{x}^*) &= 0 \end{aligned}$$

Para este problema solo seguir las especificaciones generales de la presente tarea.

Problema 3 (Wood Function):

$$\begin{aligned} f(\boldsymbol{x}) &= 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 \\ &\quad +10{,}1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19{,}8(x_2 - 1)(x_4 - 1) \\ \boldsymbol{x}^0 &= [-3, -1, -3, -1]^T \\ \boldsymbol{x}^* &= [1, 1, 1, 1]^T \\ f(\boldsymbol{x}^*) &= 0 \end{aligned}$$

Para este problema solo seguir las especificaciones generales de la presente tarea.

Detalles de Implementación

La función **gradientdescent** debe recibir los siguientes parámetros:

a) x_0: el punto inicial.
b) $mxitr$: el número máximo de iteraciones.
c) tol_g: la tolerancia para la norma del gradiente.
d) tol_x: la tolerancia para el error relativo de las x.
e) tol_f: la tolerancia para el error relativo de las evaluaciones de f
f) f: la función objetivo
g) g: el gradiente
h) H: el hessiano de f (Nota: es un parametro opcional)
i) msg: un string.

Es decir, a la función **gradientdescent** se le pasará como argumento el punto inicial, las tolerancias para los criterios de paro, y se le pasara como argumentos las funciones: f, g y H que corresponden a la función objetivo, la función gradiente y la función Hessiana. Además, msg será un *string*, el cual tomará las siguientes opciones: $msg =' StepHess'$, $msg =' Backtracking'$ y $msg =' StepFijo'$ para referirse a el método cuando actualize el tamaño de paso vía cálculo del Hessiano, Backtracking y tamaño de paso fijo, respectivamente. Por tanto la función **gradientdescent** tendría la forma:

def `gradientdescent`(x_0, $mxitr$, tol_g,tol_x, tol_f, f, g, H, msg, *args):

En el caso de $msg =' StepFijo'$ se le pasaría el valor del tamaño de paso fijo deseado como un argumento extra (por esto el uso del puntero *args*), y si lo desean pueden pasar más parámetros a la función **gradientdescent** por medio de el puntero *args*. Los criterios de paro a utilizar serán: Número máximo de iteraciones K, tol_x (para el criterio $\frac{||x_{k+1}-x_k||}{\max\{1,||x_k||\}} < tol_x$), tol_f (para el criterio $\frac{|f(x_{k+1})-f(x_k)|}{\max\{1,|f(x_k)|\}} < tol_f$), y tol_g (para el criterio $||\nabla f(x_k)|| < tol_g$).