OPTIMIZATION. HOMEWORK 6

OSCAR DALMAU

Conjunto de Problemas

(1) Implementar el método del gradiente conjugado.
(2) Con la implementación anterior, resolver el siguiente problema de optimización cuadrática:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i,j} \left[(x_{i,j} - g_{i,j})^2 + \lambda \sum_{(l,m)\in\Omega_{i,j}} (x_{i,j} - x_{l,m})^2 \right] \tag{1}$$

donde $\lambda > 0$ es un parámetro de regularización dado; g es la función a la cual se le desea suavizar (o filtrar el ruido); $\Omega_{i,j}$ es el conjunto de índices formados por posiciones vecinas a (i, j), es decir $\Omega_{i,j} = \{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$. Observe que si la funcion original es de tamaño $m \times n$ entonces funcion suavizada sera $\mathbf{x}^* = [x_{i,j}]_{1\leq i\leq m, 1\leq i\leq n}$ donde $\mathbf{x}^*$ es el minimizador del problema.

Especificaciones

Resolver el problema (1) para los valores de $\lambda \in \{1, 100, 1000\}$. Para cada uno de estos valores de λ, reportar la gráfica de las iteraciones vs la función objetivo, también deberán reportar la gráfica de las iteraciones vs la norma del gradiente. Además mostrar la funcion original original y la funcion suavizada $\mathbf{x}^* = [x_{i,j}]_{i,j}$

La función debera recibir los siguientes parámetros:

(1) x_0: el punto inicial.
(2) tol_g: la tolerancia para la norma del gradiente.
(3) f: la función objetivo
(4) $grad$: el gradiente
(5) g: la función g del modelo de optimización (1), por ejemplo, puede usar el script en Listing 1 para generar una funcion, o pudiera definir cualquier otra funcion, o incluso podria usar una imagen en grises.

LISTING 1. Ejemplo de funcion 'g' para el problema de optimización (1)

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
n = 128
x = y = np.linspace(-1,1, n)
x, y = np.meshgrid(x,y)
g = x**2 + y**2 + 0.1*np.random.randn(*x.shape)
g = (g-g.min())/(g.max()-g.min())
ax.plot_surface(x,y,g)
```