

Métodos de Optimización Lab

I. Optimización (unconstrained)

1. Haz un código simple para el gradient descent

def GradDesc(grad,x0,lamda,maxiter,xtol):

 dado un valor inicial x0, moverse en la dirección del gradiente

 El código termina cuando haya convergido, $\text{abs}(x(n+1)-x(n)) < \text{xtol}=0.0001$

 o bien se hayan cumplido maxiter=1000 iteraciones y lambda=0.1

2. Encuentra el mínimo de una parábola $y=4x^2+-8x+5$ usando este código

3. Haz un plot de la parábola (o función) y del camino que va siguiendo el algoritmo, $x1, f(x1)$ (hay que guardar en cada iteración el valor)

4. Prueba distintos valores de lambda

5. Minimiza la función con dos variables $y=x^2+xy+y^2$

6. Compara los resultados con una de las funciones de python

Optimización Lineal

Ejercicio 2.

Soluciona el siguiente problema:

1. de manera geométrica (y de paso repasamos los gráficos)

2. Usando `scipy.optimize.linprog`

$$\begin{array}{ll}\text{minimiza} & z = -x_1 - 2x_2 \\ \text{sujeto a} & -2x_1 + x_2 \leq 2 \\ & -x_1 + x_2 \leq 3 \\ & x_1 \leq 3 \\ & x_1, x_2 \geq 0\end{array}$$

3. Minimiza ahora la ecuación no lineal :

$$z = -2x_1x_2 + x_2^2$$

con las mismas condiciones

Modelizar

Ejercicio 3.

1. Define este polinomio de orden 3 $pol = -x^3 + 3x^2 + 2x + 10$ usando `numpy.poly1d` o definiendo la función

2. Genera un set de 20 datos con un ruido Gaussiano con dispersion=25

```
np.random.randn()
```

```
x=np.linspace(-10,10,20)
```

```
y=pol(x)+noise
```

3. Obtener valores de las constantes, con `polyfit`. Hacer plot de los datos con ruido y del ajuste (con leyenda)

5. Calcular residuos $\text{suma}(y_{\text{ruidos}} - y_{\text{estimado}})^2$

6. Hacer mismo ejercicio con `optimize.minimize` con el método que queráis usando como función objetivo el error mínimo cuadrático

Modelizar

Ejercicio 4.

1. Dada la función: $y = ae^{-b \sin(fx + \phi)}$

2. Generamos set de datos con ruido:

true_params=[3, 2, 1, np.pi/4]:

x = np.linspace(0, 2*np.pi, 25)

y = function(x, *true_params)

noise = exact + 0.3*np.random.randn()

y=y+noise

usa los parametros : true_params=[3, 2, 1, np.pi/4]:

3. Encuentra parámetros que minimizan residuos con curve_fit y comparalos con los de verdad

4. Haz plot comparando datos y ajuste

5. Probad distintos, puntos iniciales, incrementad nivel de ruido, poner cotas (bounds)

6. Hacer mismo ejercicio con optimize.minimize con el metodo que queráis usando como función objetivo el error mínimo cuadrático

Modelizar

Ejercicio 5.

1. Carga los datos dataSin.txt
2. Busca el polinomio de grado mínimo que da un buen ajuste para estos datos. (Calcula mse para cada polinomio ajustado)
3. Ajusta con `curve_fit` estos datos a la curva $y = A \sin(bx)$
4. Tengo yo los resultados

Optimización No Lineal

1. Using the minimum function, compute the minimum of the scalar function $f(x, y) = (x - 1)^2 + (y - 2)^2 - 3x + 2y + 1$ in two ways: with and without providing the Jacobian.
2. Try to maximize the function $f(x, y) = xy$ subject to the equality constraints

$$2x^2 + y^2 = 1$$

$$x^2 + 3y^2 = 1$$