Ejercicios

Optimization and Modelization Lab

Optimization:

1. Jugar con ejemplo dieta, cambiando restricciones (optimizationExmple.ipynb)

2. (Pensar y escribir ecuaciones) y resolver con python problema 2.

3 . Resolver gráficamente Problema 2

4. Resolver con python problema 3.

PROGRAMACiÓN LINEAL. Problema 2 (ecuaciones en presentación)

Lineas de producción:
Tenemos 2 líneas de producción y 3 productos. La empresa se ha comprometido a proveer 600 refrescos, 400 zumos y 1000 botellines de agua cada día a una distribuidora.

Línea	Coste hora (€)	Refresco (u.)	Zumo (u.)	Agua (u.)
1	20	60	30	40
2	6	10	10	60

Loss function
- Minimizar coste

Objetivo
- Encontrar número de horas que trabaja cada línia para conseguir minimizar el coste manteniendo los compromisos.

Optimización con ligaduras o restricciones.
PROGRAMACiÓN LINEAL. Problema 2 ecuaciones

Variables:

horas que tengo activa cada línia,

$$h_1, h_2$$

Función objetivo:

$$f = 20h_1 + 6h_2$$

Ligaduras:

$$60h_1 + 10h_2 > 600$$
$$30h_1 + 10h_2 > 400$$
$$40h_1 + 60h_2 > 1000$$
$$h_1, h_2 <= 24$$
$$h_1, h_2 >= 0$$

PROGRAMACiÓN CUADRÁTICA.
Problema 3

$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & z = -2x_1x_2 + x_2^2 \\
\text{subject to} \quad & -2x_1 + x_2 \leq 2, \\
& -x_1 + x_2 \leq 3 \\
& x_1 \leq 3 \\
& x_1, x_2 \geq 0
\end{aligned}$$

Modelization:

Ejercicio 1:

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

2. Simula unos datos (~alrededor de 20), siguiendo ejemplo del notebook (ModelisationExample.ipynb)
   - x entre -10 y 10
   - y --> pol(x) + un ruido Gaussiano con un ruido Gaussiano con dispersion=25

3. Haz ajuste con polyfit, compara valores que obtienes con los reales.

4. Haz plot de los datos y del ajuste (con leyenda)

5. Calcular error cuádratico promedio (mse)

Modelization:

Ejercicio 2:

1. Carga los datos en Data_exp.txt (np.loadtxt())

2. Asumiendo que siguen esta función hacer ajuste usando curve_fit

$$y = ae^{-b\sin(fx+\phi)}$$

3. Haz plot de los datos y del ajuste (con leyenda)

4. Busca el polinomio de grado mínimo que da un buen ajuste para estos datos. (Calcula mse para cada polinomio ajustado)

5. Compara los resultados usando el polinomio y la curva real (apartado 2.)

6. Carga los datos extendidos: Data_exp_large.txt y compáralos con los ajustes hechos ya.
(Con un ajuste polinómico no se puede generalizar, aunque es muy útil para interpolar)

Optimization Solutions:

Problema 2 :
h_1= 6,66 horas, h_2=20 horas, Coste total =253 €

Problema 4:
x_1= 17.7, x_2= 7.6; Min Func = -210

Modelization Solutions:

Ejercicio 1 :
mse ~25

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

PROGRAMACiÓN LINEAL.
Problema auxiliar

<u>Interese préstamos bancarios:</u>
- Primera hipoteca 14%
- Segunda hipoteca 20%
- Rehabilitación 20%
- Crédito personal 10%

<u>Políticas del banco:</u>
- No puede prestar más de 250 millones de euros
- Primeras hipotécas tienen que ser al menos un 55% de todas las hipotécas y como mínimo el 25% de todos los préstamos.
- Las segundas hipotecas no pueden exceder el 25% de todos los prestamos.
- El total de los intereses cobrados no superaran el 15% del total prestado.

<u>Objetivo</u>
- Maximizar ingresos debido a los intereses cobrados teniendo en cuenta las políticas bancarias

PROGRAMACiÓN NO-LINEAL.
Problema auxiliar

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

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