



UNIVERSIDAD
CATÓLICA DE
TEMUCO

FACULTAD DE
INGENIERÍA

Grupo 7

Maximiliano Gonzalez

José Moya

Fernando Torres

Eduardo Lobos

Abril 2021

1. Introducción

Durante las semanas anteriores, se estuvo realizando la actividad explicada por medio de los vídeos y el guión proporcionados por el profesor. En esta actividad, se plantea el objetivo el cual consiste en optimizar modelos de disolución de fármacos, en específico la función que determina el costo aplicando cada modelo, al ser costo buscamos minimizar la función. Para esto se requiere de crear un algoritmo en python, matlab o algún otro lenguaje de programación, donde se implementen métodos de optimización pertinentes para esta actividad.

Los modelos a utilizar son el modelo de Weibull y el modelo de Korsmeyer-Peppas, los cuales tienen la forma:

$$u(t) = 1 - e^{-at}$$

$$u(t) = at^b$$

Y la función de costo quedaría expresada así:

$$\sum_{i=1}^n (y(p; x_i) - y_i)^2$$

Y aplicando los modelos que tenemos la sumatoria quedaría así:

$$\sum_{i=1}^n (u(t) - y_i)^2$$

Esto cambiara dependiendo de que modelo utilicemos. También se nos entrega una tabla de datos que serán nuestros "x.y" la cual da estos datos:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
y	0.2	0.3	0.45	0.55	0.6	0.7	0.75	0.8	0.8	0.8

2. Desarrollo

Para empezar debemos calcular la derivada parcial de los modelos

$$u(t) = 1 - e^{-at}$$

$$u(t) = at^b$$

$$\nabla(u(t)) = (ae^{-at}) \tag{1}$$

$$\nabla(u(t)) = (abt^{b-1}) \tag{2}$$

3. Código Inicial

Para realizar nuestra comprobación con los datos proporcionados en la actividad creamos un algoritmo simple, el cual utiliza ambos modelos originales, los datos de prueba entregados en la guía y los valores de los parámetros A y B de manera aleatoria, de esta manera, se ejecuta la sumatoria con estos parámetros aleatorios y los otros datos durante 100 repeticiones, de donde se obtiene la sumatoria con el menor valor de todos. Esta sumatoria se ve reflejada en la consola de la ejecución del programa, en conjunto con lo anterior se genera un archivo JSON el cual contiene todos los valores de estas sumatorias, en conjunto con los parámetros utilizados.

```
1 import math
2 import random
3 import json
4 def M1(a,t):
5     Y=1-pow(math.e,-(a*t))
6     return Y
7
8 def M2(a,b,t):
9     Y=pow((a*t),b)
10    return Y
11
12 def Sumatoria(Funcion,T,U,Z):
13     if Funcion=="M1":
14         f=[]
15         ft={}
16         ft['Puntos']=[]
17         for j in range(0,len(Z)):
18             for i in range(0,len(T)):
19                 f.append(pow(M2(Z[j][0],Z[j][1],T[i])-U[i],2))
20             ft['Puntos'].append({
21                 'P':str(Z[j][0])+' '+str(Z[j][1]),
22                 'Valores Sumatoria':f,
23                 'Valor Minimo de Sumatoria':min(f),
24                 'Suma':sum(f)
25             })
26         f=[]
27         return ft
28     elif Funcion=="M2":
29         f=[]
30         ft={}
31         ft['Puntos']=[]
32         for j in range(0,len(Z)):
33             for i in range(0,len(T)):
34                 f.append(pow(M1(Z[j],T[i])-U[i],2))
35             ft['Puntos'].append({
36                 'P':str(Z[j]),
37                 'Valores Sumatoria':f,
38                 'Valor Minimo de Sumatoria':min(f),
39                 'Suma':sum(f)
40             })
41         f=[]
42         return ft
43     else:
```

```

44         print("Funci n Inv lida")
45
46
47 T=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
48 U=[0.2,0.3,0.45,0.55,0.6,0.7,0.75,0.8,0.8,0.8]
49 P1=[]
50 P2=[]
51 for i in range(1,100):
52     P1.append([random.randint(1,50),random.randint(1,50)])
53     P2.append(random.randint(1,50))
54
55 print("Generando datos")
56 F1=Sumatoria("M1",T,U,P1)
57 F2=Sumatoria("M2",T,U,P2)
58
59 print("Escribiendo JSON")
60 with open('M1.json', 'w') as file:
61     json.dump(F1, file, indent=4)
62
63 with open('M2.json', 'w') as file:
64     json.dump(F2, file, indent=4)
65
66 print("Completado!")
67
68 print("Buscando m nimos")
69
70 with open('M1.json') as file:
71     data = json.load(file)
72     Lista=[]
73     for Puntos in data['Puntos']:
74         Lista.append([str(Puntos['P']),Puntos['Suma']])
75     Min=[]
76     for D in Lista:
77         Min.append(D[1])
78     MinVal=min(Min)
79     for D in Lista:
80         if MinVal==D[1]:
81             PuntoMin=D[0]
82             break
83     for Puntos in data['Puntos']:
84         if str(Puntos['P'])==PuntoMin:
85             print("\n")
86             print("Modelo 1")
87             print("P:",Puntos['P'])
88             print("Valores Sumatoria:",Puntos['Valores Sumatoria'])
89             print("Valor Minimo de Sumatoria:",Puntos['Valor Minimo de
Sumatoria'])
90             print("Suma:",Puntos['Suma'])
91             break
92
93 with open('M2.json') as file:
94     data = json.load(file)
95     Lista=[]
96     for Puntos in data['Puntos']:
97         Lista.append([str(Puntos['P']),Puntos['Suma']])
98     Min=[]
99     for D in Lista:

```

```

100     Min.append(D[1])
101     MinVal=min(Min)
102     for D in Lista:
103         if MinVal==D[1]:
104             PuntoMin=D[0]
105             break
106     for Puntos in data['Puntos']:
107         if str(Puntos['P'])==PuntoMin:
108             print("\n")
109             print("Modelo 2")
110             print("P:",Puntos['P'])
111             print("Valores Sumatoria:",Puntos['Valores Sumatoria'])
112             print("Valor Minimo de Sumatoria:",Puntos['Valor Minimo de
Sumatoria'])
113             print("Suma:",Puntos['Suma'])
114             break

```