

Actividad 4: Ajuste de Datos

Paulina Valenzuela Coronado

February 15, 2016

1 Introducción

En esta actividad se utilizó Python para realizar un ajuste de datos con el método de mínimos cuadrados. El método de mínimos cuadrados es una técnica de análisis numérico enmarcada dentro de la optimización matemática, en la que, dados un conjunto de pares ordenados: variable independiente, variable dependiente, y una familia de funciones, se intenta encontrar la función continua, dentro de dicha familia, que mejor se aproxime a los datos con un "mejor ajuste", de acuerdo con el criterio de mínimo error cuadrático.[1]

En su forma más simple, intenta minimizar la suma de cuadrados de las diferencias en las ordenadas (llamadas residuos) entre los puntos generados por la función elegida y los correspondientes valores en los datos.

Para esta actividad se ajustaron dos colecciones de datos, a cada uno se le ajustó de manera distinta, utilizando herramientas del paquete *Spicy* de Phyton. *scipy.optimize.leastsq* encuentra el conjunto de parámetros que minimizan la función de error. Este comienza a partir de una primera conjetura y trata de minimizar la función de error cada que entran los datos proporcionados, y así devuelve la lista de los parámetros que mejor se ajusten a los datos [2].

2 Código

- **Ajuste Lineal**

Para este ajuste se utilizó una colección de datos de **Temperaturas de invierno en Nueva York**.

```
import numpy as np
from numpy import pi, r_
import matplotlib.pyplot as plt
from scipy import optimize

datos = np.loadtxt('temperatura')
x=datos[:,0]
y=datos[:,1]
print x
print y

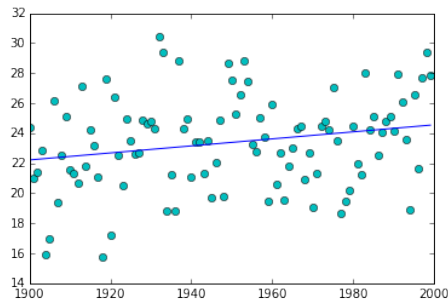
fitfunc = lambda p, x: p[0] + p[1]*x # Target function
errfunc = lambda p, x, y: fitfunc(p, x) - y # Distance to the target function
```

```

p0 = [-15., 0.8] # Initial guess for the parameters
p1, success = optimize.leastsq(errfunc, p0[:], args=(x, y))

time = np.linspace(x.min(), x.max(), 100)
plt.plot(x, y, "co", time, fitfunc(p1, time), "b-") # Plot of the data and the fit

```



- **Ajuste Exponencial**

Para este ajuste se utilizó una colección de datos de **Presion atmosférica vs. altitud**.

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit

#Leyendo el archivo con los datos
datos = np.loadtxt('Presion')
x=datos[:,0]
y=datos[:,1]
print x
print y

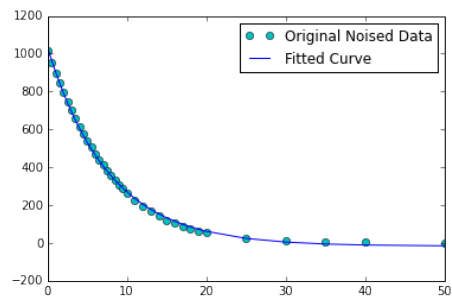
def func(x, a, b, c):
    return a * np.exp(-b * x) + c

yn = y + 0.2*np.random.normal(size=len(x))

popt, pcov = curve_fit(func, x, yn)

plt.figure()
plt.plot(x, yn, 'co', label="Original Noised Data")$
plt.plot(x, func(x, *popt), 'b-', label="Fitted Curve")
plt.legend()
plt.show()

```



References

- [1] MÍNIMOS CUADRADOS <https://es.wikipedia.org/wiki/M>
- [2] SCIPY.OPTIMIZE.LEASTSQ <http://goo.gl/giZ7rO>