

Actividad 4: Ajuste de datos; mínimos cuadrados

Martín Alejandro Paredes Sosa

Febrero 2016

1. Introducción

En el área de la estadística, hay un campo denominado análisis de regresión cuya misión es poder obtener modelos matemáticos, como las funciones, para diversos comportamientos de fenómenos observados. Tras tener una colección de datos, estos se analizan y con diferentes métodos, se llega a facilitar el estudio del fenómeno con funciones conocidas, como las rectas, parábolas o exponenciales.

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 (un “mejor ajuste”), de acuerdo con el criterio de mínimo error cuadrático [1].

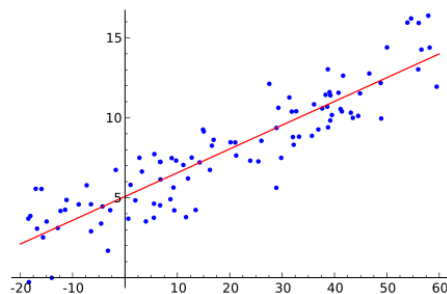


Figura 1: Ejemplo de una regresión lineal.[2].

La técnica de los mínimos cuadrados es parte de ambas secciones en el análisis de la regresión, pues se puede utilizar tanto en la regresión lineal como en el no lineal. En la actividad de hoy, se utilizaron ambos métodos para poder ajustar funciones del tipo lineal y exponencial a dos conjuntos de datos distintos con una herramienta que nos proporciona el paquete Scipy de Python.

El método *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 [3].

2. Actividad

En esta practica se realizara un ajuste lineal de temperaturas medias de invierno en Nueva York[4] y un ajuste exponencial de la presión atmosférica con respecto a la altura[5], utilizando los paquetes de *Scipy: curve_fit; leastsq*.

2.1. Programa 1: AjLineal.py

En este programa se lee el documento con los datos de temperaturas medias de invierno en Nueva York[4], y se realiza un ajuste lineal.

Código: AjLineal.py

```
#importando librerias
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import leastsq

#Obtencion de datos
datos = np.loadtxt('TNY.txt')

#Separacion de los datos para un mejor manejo
x1 = datos[:,0]
```

```

y1 = datos[:,1]

#Ajuste Lineal

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., -1.] # Initial guess for the parameters
p1, success = leastsq(errfunc, p0[:], args=(x1, y1))

#Genero datos y grafico
time = np.linspace(x1.min(), x1.max(), 100)
plt.plot(x1, y1, "go", time, fitfunc(p1, time), "b-") # Plot of the data and the f

#Propiedades de la Grafica
plt.title("Temperatura Nueva York en invierno")
plt.xlabel("Año")
plt.ylabel("Temperatura")

plt.show()

```

Lo que resulta de este código es lo siguiente:

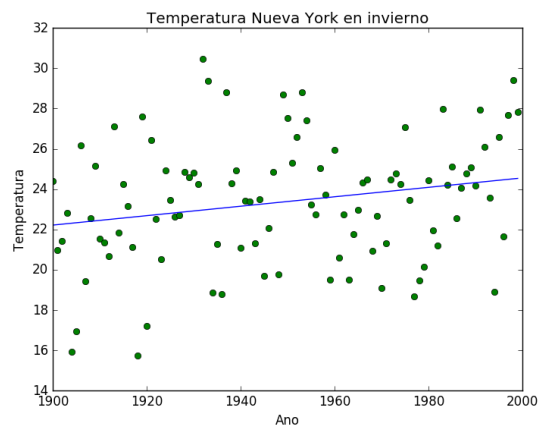


Figura 2: Ajuste Lineal, Temperatura Nueva York en invierno

2.2. Programa 1: AjExp.py

En este programa se lee el documento con los datos de la presión atmosférica con respecto a la altura[5], y se realiza un ajuste exponencial.

Código: AjExp.py

```
#importando librerias
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import curve_fit

#Obtencion de datos
datos = np.loadtxt('PvsA.txt')

#Separacion de los datos para un mejor manejo
x1 = datos[:,0]
y1 = datos[:,1]

#Definiendo la forma de la función a ajustar
# $y=c*e^{(-a*x)}+b$ 
def f(x,a,b,c):
    return c * np.exp(-a * x) + b

#Optimizacion de la curva
popt, pcov = curve_fit(f,x1,y1)

#Genero datos y grafico

plt.plot(x1, y1, "go", label='Datos')
plt.plot( x1, f(x1, *popt), "b-", label='Ajuste Exponencial')

#Propiedades de la Grafica
plt.grid()
plt.legend()
plt.title("Atmospheric pressure vs. altitude")
plt.xlabel("Altitude")
plt.ylabel("Pressure")
```

```
plt.show()
```

Lo que resulta de este código es lo siguiente:

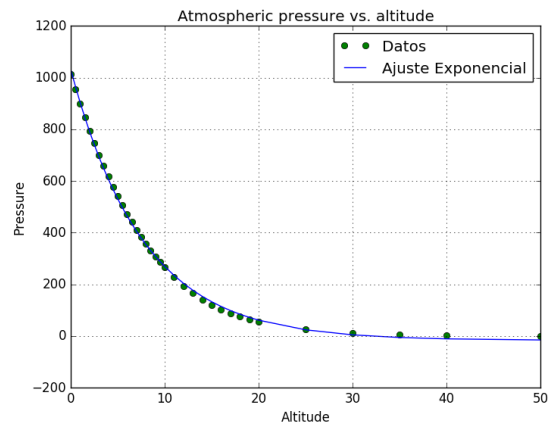


Figura 3: Ajuste Exponencial, Presión Atmosférica con Respecto a la Altura

Referencias

- [1] Wikipedia, *Mínimos cuadrados*. Recuperado de:
https://es.wikipedia.org/wiki/M%C3%ADnimos_cuadrados
- [2] Lizárraga, C. (2016) *Actividad 4 (2016-1)*. Recuperado de:
[http://computacional1.pbworks.com/w/page/105016164/Actividad %204 %20\(2016-1\)](http://computacional1.pbworks.com/w/page/105016164/Actividad%204%20(2016-1))
- [3] Qiku programación, (2013) *Cómo utilizar leastsq función desde scipy.optimize en python*. Recuperado de: <http://goo.gl/giZ7rO>
- [4] Quantitative Environmental Learning Project, (s.f.) *New York's Winter Mean Temperature* Recuperado de:
[http://www.seattlecentral.edu/qelp/sets/048/048.html# Top](http://www.seattlecentral.edu/qelp/sets/048/048.html#Top)
- [5] Quantitative Environmental Learning Project, (s.f.) *Earth's Standard Atmosphere*. Recuperado de:
<http://www.seattlecentral.edu/qelp/sets/024/024.html>
- [6] *Regresión lineal*. (s.f.) Recuperado de:
<http://www.sc.ehu.es/sbweb/fisica/cursoJava/numerico/regresion/regresion.htm>
- [7] *Regresión exponencial*. (s.f.) Recuperado de
<https://www.uv.es/ceaces/base/regresion/exponenci.htm>
- [8] Scipy (2016), *leastsq*. Recuperado de:
<http://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.leastsq.html#scipy.optimize.leastsq>
- [9] Scipy (2016), *curve_fit*. Recuperado de:
http://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html