

Práctica 0: Python

Aprendizaje Automático y Big Data



4 de octubre de 2018

Félix VIllar y Víctor ramos

Universidad Complutense de Madrid

***Versión 1: Forma Iterativa***

La primera parte de la práctica consistía en realizar el cálculo de una integral mediante el método de Montecarlo.

***CÓDIGO:***

import numpy as np

from scipy.integrate import quad

import matplotlib.pyplot as plt

def integra\_mc(fun,a,b,num\_puntos=1000):

losx =[]

losy =[]

vx =[]

vy =[]

mx=0

nd=0

for funcion in range(num\_puntos): #Representacion de la funcion

fx = np.random.uniform(a,b)

fy = fun(fx)

vx.append(fx)

vy.append(fy)

if(mx < fy):

mx = fy

for valor in range(num\_puntos): #numeros aleatorios x e y dispersos entre a y b

x = np.random.uniform(a,b)

y = np.random.uniform(0,mx)

z = fun(x)

losx.append(x)

losy.append(y)

if(z > y):

nd+=1

plt.figure()

plt.plot(losx,losy,"x",vx,vy,'o')

plt.show()

resul = (nd/num\_puntos)\*(b-a)\*mx

print(resul)

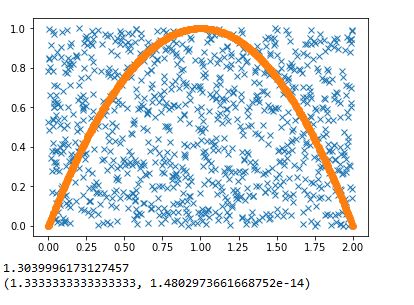
r = quad(fun,a,b)

print(r)

x= lambda a: -(a\*a) +2\*a

integra\_mc(x,0,2)

***EJEMPLO***



P.D. El primer número que se muestra es el calculado por nuestro algoritmo y el segundo es el facilitado por integrate.quad()

***VERSIÓN 2: VECTORIZACIÓN***

La segunda parte de la práctica consistía en hacer la misma acción pero utilizando técnicas de broadcasting y vectorización.

***CÓDIGO:***

import numpy as np

from scipy.integrate import quad

import matplotlib.pyplot as plt

def integra\_mc(fun,a,b,num\_puntos=10000000):

mx=0

fx = np.random.rand(num\_puntos)\*(b-a) + a

fy = fun(fx)

mx = np.amax(fy)

x = np.random.rand(num\_puntos)\*(b-a) + a

y = np.random.rand(num\_puntos)\*mx

z = fun(x)

nd = np.sum(z>y)

#GRAFICA CON LOS DATOS

plt.figure()

plt.plot(x,y,"x",fx,fy,'o')

plt.show()

resul = (nd/num\_puntos)\*(b-a)\*mx

print(resul)

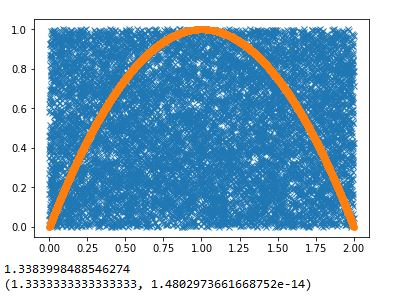
r = quad(fun,a,b)

print(r)

x= lambda a: -(a\*a) +2\*a

integra\_mc(x,0,2)

***EJEMPLO***



P.D. El primer número que se muestra es el calculado por nuestro algoritmo y el segundo es el facilitado por integrate.quad()