

Compte-rendu TP sur l'intégration numérique

Tom Chauvel

Contents

= Generate data

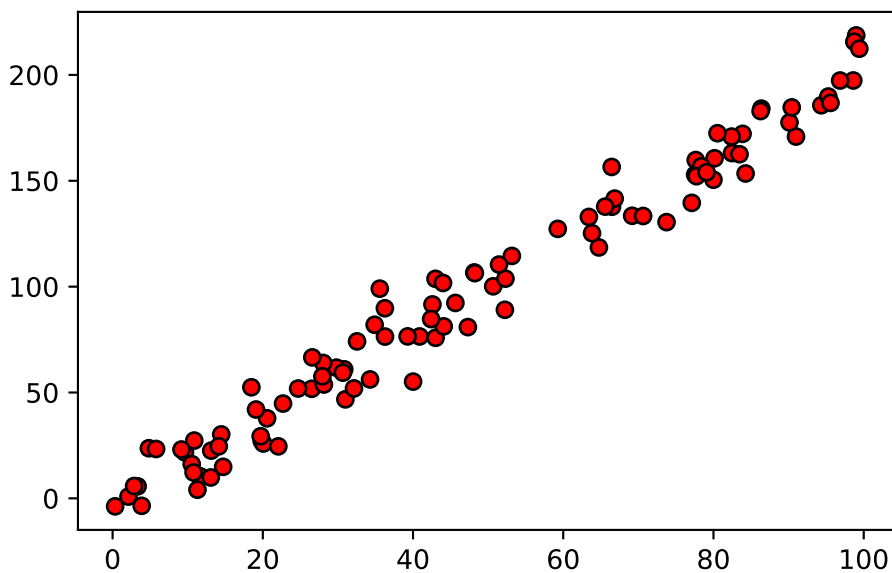
```
import numpy as np
import matplotlib.pyplot as plt

# Number of data points
N = 100

X = np.random.uniform(low=0, high=100, size=N)

# Making y = 2x + 1 + some gaussian or normal noise (assumption of linear
# regression itself)
Y = 2 * X + 1 + np.random.normal(scale=10, size=N)

# Plotting the data to see if it looks linear
plt.scatter(X, Y, edgecolors='black', color="red")
plt.show()
```



les paramètres d'un modèle linéaire sont la pente (w_0) et l'ordonnée à l'origine (w_1).

= Loss function

```
def SSR(w,x,y):
    N = len(y)
    s = 0

    lw = len(w)
    fw = lambda x : sum( w[i]*x**(lw-i-1) for i in range(lw) )

    for i in range(0,N):
        s += ( fw( x[i] ) - y[i] )**2
    return s

def MSE(w,x,y):
    N = len(y)
    return SSR(w,x,y)/N
```

```
print(SSR([2,1],X,Y))
print(MSE([2,1],X,Y))
```

```
10611.321361525486
106.11321361525486
```

= Gradient descent

```
def gradient(w,x,y):
    lw = len(w)
    fw = lambda x : sum( w[i]*x**(lw-i-1) for i in range(lw) )

    l = ( fw(x) - y )

    grad_w = np.array([ sum(2 * x ** (lw-i-1) * l) for i in range(lw) ])

    return grad_w

gradient([2,1],X,Y)
```

```
array([1281.0548388 , 227.52501592])
```

```
def descenteGradient(w,x,y,μ,n):
    for _ in range(n):
        grad_w = gradient(w,x,y)
        w -= μ * grad_w
    return w
```

```
def descenteGradient2(w,x,y,μ,ε):
    normeE = np.inf
    while normeE > ε:
        grad_w = gradient(w,x,y)
        w -= μ * grad_w
        normeE = np.abs(np.max(grad_w))
    return w

print( descenteGradient([0,0],X,Y,10**(-6),100) )
print( descenteGradient2([0,0],X,Y,10**(-6),10**(-1)) )
```

```
[2.01287455 0.01512762]
[ 2.0544249 -2.728125 ]
```

= Experiment

```
X = np.random.uniform(low=0, high=100, size=N)
Y = 2 * X + 1 # + np.random.normal(scale=10, size=N)

print( descenteGradient([0,0],X,Y,10**(-3),100) )
print( descenteGradient2([0,0],X,Y,10**(-3),10**(-1)) )
```

```
[-3.13744424e+280 -4.74950810e+278]
[nan nan]
```

```
/tmp/ipykernel_738/3115440749.py:7: RuntimeWarning: overflow encountered in
scalar add
    grad_w = np.array([ sum(2 * x ** (lw-i-1) * l) for i in range(lw) ])
/tmp/ipykernel_738/3171002804.py:11: RuntimeWarning: invalid value encountered
in subtract
    w -= μ * grad_w
```