Búsqueda lineal imprecisa

Penadillo Lazares Wenses Johan

UNIVERSIDAD
NACIONAL DE
INGENIERÍA

8 de noviembre de 2021

Índice

Introducción

$$S(x,d) = \{y : y = x + \alpha d, \alpha \geq 0, f(y) = \min_{0 \leq \alpha \leq \infty} f(x + \alpha d)\}$$

Percentage Test

Se elige una constante c talque $0 < c < 1$, un valor usual es $c = 0.10$. $|\alpha - \bar{\alpha}| \leq c\bar{\alpha}$, donde $\bar{\alpha}$ es el valor mínimo real.

Armijo's Rule

$$\phi(\alpha) = f(x_k + \alpha d_k)$$

Teniendo en consideración $\phi(0) + \varepsilon\phi'(0)\alpha$, para un valor fijo de ε, $0 < \varepsilon < 1$.
Un α sera considerado no muy grande si:

$$\phi(\alpha) \leq \phi(0) + \varepsilon\phi'(0)\alpha$$

Y un α sera considerado no muy pequeño, $\eta > 1$ si:

$$\phi(\eta\alpha) > \phi(0) + \varepsilon\phi'(0)\eta\alpha$$

Goldstein Test

ε, $0 < \varepsilon < \frac{1}{2}$.
Un α sera considerado no muy grande si:

$$\phi(\alpha) \leq \phi(0) + \varepsilon\phi'(0)\alpha$$

Y un α sera considerado no muy pequeño si:

$$\phi(\alpha) > \phi(0) + (1-\varepsilon)\phi'(0)\alpha$$

$x_{k+1} = x_k + \alpha d_k$

$$\varepsilon \leq \frac{f(x_{k+1}) - f(x_k)}{\alpha \nabla f(x_k) d_k} \leq 1 - \varepsilon$$

Wolfe Test

ε, $0 < \varepsilon < \frac{1}{2}$.

$$\phi'(\alpha) \geq (1 - \varepsilon)\phi'(0)$$

Backtracking

Iniciaremos con $\alpha = 1$, $\eta > 1$, $\varepsilon < 1$ (usualmente $\varepsilon < 0.5$). El criterio de parada sera el primero del criterio de Armijo o Goldstein. $(\phi(\alpha) = f(x_k + \alpha d_k)$ talque α cumpla $\phi(\alpha) \leq \phi(0) + \varepsilon\phi'(0)\alpha$. Si no se comple entonces α sera reducido en un factor de $\frac{1}{\eta}$. Esto es $\alpha_{new} = \frac{\alpha_{old}}{\eta}$. Valores usuales para η son 1.1 o 1.2.

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

def f(x):
  return x**2+20*x

def df(x):
  return 2*x+20

def line_search(f, df, x, alpha, d, eps = 1.0e-8):
  e = 0.3
  n = 1.2
  for i in range(600):
    while(f(x+alpha*d) > f(x)+e*df(x)*alpha):
      alpha = alpha/n
    print("alpha_"+str(i)+" = "+str(alpha))
    xk = x + alpha*d
    print("x_"+str(i)+" = "+str(x))
    if(abs(xk - x) < eps):
      return xk
    x = xk


```

```
x = -20.0
alpha = 1.0
d = 2.0

X = np.arange(-30,30,0.1)
Y = f(X)
plt.plot(X,Y)
plt.xlabel('x')
plt.ylabel('y')
plt.show()

print(line_search(f,df,x,alpha,d))
```