Numerical Optimization
Constrained & Unconstrained Optimization Algorithms for Machine Learning models

Donato Meoli

January 17, 2020

1 Unconstrained Optimization

Algorithm 1 Pseudocode for backtracking line search.

procedure BLS$(\varphi, \varphi', \alpha, m_1, \tau)$
  while $(\varphi(\alpha) > \varphi(0) + m_1 \alpha \varphi'(0))$ **do**
    $\alpha \leftarrow \tau\, \alpha$ ▷ $\tau < 1$

Algorithm 2 Pseudocode for quadratic functions local minimum detection.

procedure SDQ(Q, q, x, ε)
  while $(\|\nabla f(x)\| > \varepsilon)$ **do**
    $d \leftarrow -\nabla f(x)$
    $\alpha \leftarrow \|d\|^2 / (d^T Q d)$
    $x \leftarrow x + \alpha d$

Algorithm 3 Pseudocode for non-quadratic functions local minimum detection.

procedure SDG(f, x, ε)
  while $(\|\nabla f(x)\| > \varepsilon)$ **do**
    $d \leftarrow -\nabla f(x)$
    $\alpha \leftarrow \mathrm{AWLS}(f(x + \alpha d))$
    $x \leftarrow x + \alpha d$

Algorithm 4 Pseudocode for conjugate gradient method for quadratic functions.

procedure CGQ(Q, q, x, ε)
  $d^- \leftarrow 0$
  while $(\|\nabla f(x)\| > \varepsilon)$ **do**
    if $(d^- = 0)$ **then**
      $d \leftarrow -\nabla f(x)$
    else
      $\beta = (\nabla f(x)^T Q d^-)/(d^{-T} Q d^-)$
      $d \leftarrow -\nabla f(x) + \beta d^-$
    $\alpha \leftarrow (\nabla f(x)^T d)/(d^T Q d)$
    $x \leftarrow x + \alpha d$
    $d^- \leftarrow d$

Algorithm 5 Pseudocode for conjugate gradient method for arbitrary functions.
We have three different formulas for β, which coincide in the quadratic case:

$\triangleright$ Polak-Ribière: $\beta = (\nabla f(x)^T (\nabla f(x) - \nabla f(x^-)))/\|\nabla f(x^-)\|^2$

$\triangleright$ Hestenes-Stiefel: $\beta = (\nabla f(x)^T (\nabla f(x) - \nabla f(x^-)))/((\nabla f(x) - \nabla f(x^-))^T d^-)$

$\triangleright$ Dai-Yuan: $\beta = \|\nabla f(x)\|^2/((\nabla f(x) - \nabla f(x^-))^T d^-)$

procedure CGA(f, x, ε)
  $\nabla f^- = 0$
  while $(\|\nabla f(x)\| > \varepsilon)$ **do**
    if $(\nabla f^- = 0)$ **then**
      $d \leftarrow -\nabla f(x)$
    else
      $\beta = \|\nabla f(x)\|^2/\|\nabla f^-\|^2$ $\triangleright$ Fletcher-Reeves
      $d \leftarrow -\nabla f(x) + \beta d^-$
    $\alpha \leftarrow \mathrm{AWLS}(f(x + \alpha d))$
    $x^- \leftarrow x$
    $x \leftarrow x + \alpha d$
    $d^- \leftarrow d$
    $\nabla f^- \leftarrow \nabla f(x)$

Algorithm 6 Pseudocode for accelerated gradient method.

```
procedure ACCG(f, x, ε)
    x₋ ← x
    γ ← 1
    repeat
        γ₋ ← γ
        γ ← ( √(4γ² + γ⁴) − γ² )/2
        β ← γ( 1 / γ₋ − 1 )
        y ← x + β( x − x₋ )
        g ← ∇f(y)
        x₋ ← x
        x ← y − ( 1 / L )g
    until (‖g‖ > ε)
```

Algorithm 7 Pseudocode for boundle method.

```
procedure PBM(f, g, x̄, m₁, ε)
    choose μ
    𝓑 ← {(x̄, f(x̄), g(x̄))}
    while ( true ) do
        x* ← argmin {f_𝓑(x) + μ‖x − x̄‖²/2}
        if (μ‖ x* − x̄ ‖₂ ≤ ε) then
            break
        if (f(x*) ≤ f(x̄) + m₁(f_𝓑(x*) − f(x̄))) then
            x̄ ← x*
            possibly decrease μ
        else
            possibly increase μ
        𝓑 ← 𝓑 ∪ ( x* , f(x*) , g(x*) )
```