BFGS

June 16, 2020

1 Hessian approximations

1.1 BFGS update

Assuming B_k is a summetric positive definite matrix, the BFGS update is defined as

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}$$

where $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ and $s_k = x_{k+1} - x_k$. Its implementation in Julia is direct.

[18]:
```
function BFGSUpdate(B, y, s)
    Bs = B*s
    return B - (Bs*Bs')/dot(s, Bs) + (y*y')/dot(s,y)
end
```

[18]:
```
BFGSUpdate (generic function with 3 methods)
```

[26]:
```
function BFGSUpdate!(B, y, s)
    Bs = B*s
    B[:,:] = B - (Bs*Bs')/dot(s, Bs) + (y*y')/dot(s,y)
end
```

[26]:
```
BFGSUpdate! (generic function with 1 method)
```

[1]:
```
using LinearAlgebra
```

[20]:
```
n = 3
y = [ 1.0 2 3]'
s = [ 0.5 0.5 0.5 ]'
```

[20]:
```
3×1 Adjoint{Float64,Array{Float64,2}}:
 0.5
 0.5
 0.5
```

[9]:
```
BFGSUpdate(I, y, s)
```

```
MethodError: no method matching BFGSUpdate(::UniformScaling{Bool}, ::
↪Array{Int64,2}, ::Array{Float64,2})

Stacktrace:

 [1] top-level scope at In[9]:1
```

[29]:
```
B = zeros(n,n)+I
```

[29]:
```
3×3 Array{Float64,2}:
 1.0  0.0  0.0
 0.0  1.0  0.0
 0.0  0.0  1.0
```

[21]:
```
BFGSUpdate(B, y, s)
```

[21]:
```
3×3 Array{Float64,2}:
 1.0       0.333333  0.666667
 0.333333  2.0       1.66667
 0.666667  1.66667   3.66667
```

[22]:
```
B
```

[22]:
```
3×3 Array{Float64,2}:
 1.0  0.0  0.0
 0.0  1.0  0.0
 0.0  0.0  1.0
```

[30]:
```
BFGSUpdate!(B, y, s)
```

[30]:
```
3×3 Array{Float64,2}:
 1.0       0.333333  0.666667
 0.333333  2.0       1.66667
 0.666667  1.66667   3.66667
```

[31]:
```
B
```

[31]:
```
3×3 Array{Float64,2}:
 1.0       0.333333  0.666667
 0.333333  2.0       1.66667
 0.666667  1.66667   3.66667
```

[33]:
```
BFGSUpdate!(B, y, s)
```

[33]:
```
3×3 Array{Float64,2}:
 1.0       0.333333  0.666667
 0.333333  2.0       1.66667
 0.666667  1.66667   3.66667
```

It is however often more convenient to work with the inverse. A technical derivation gives

$$B_{k+1}^{-1} = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) B_k^{-1} \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}$$

The corresponding Julia implementation follows.

[34]:
```
function InvBFGSUpdate_naive(invB::Matrix, y::Vector, s::Vector)
    ys = dot(y, s)
    A = I-(s*y')/ys
    return A*invB*A' + (s*s')/ys
end
```

[34]:
```
InvBFGSUpdate_naive (generic function with 1 method)
```

This implementation is however inefficient as it involves a temporary matrix. We can avoid by reorganizing the terms, recognizing that B_k^{-1} is symmetric, and that $y_k^T B_k^{-1} y_k$ and $s_k^T y_k$ are scalar. This leads to

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k^T y_k + y_k^T B_k^{-1} y_k) s_k s_k^T}{(s_k^T y_k)^2} - \frac{B_k^{-1} y_k s_k^T + s_k y_k^T B_k^{-1}}{s_k^T y_k}$$

The corresponding Julia implementation is

[41]:
```
function InvBFGSUpdate(invB::Matrix, y::Vector, s::Vector)
    ys = dot(y, s)
    invBy = invB*y
    return invB+1.0/ys*((ys+dot(y, invBy))/ys*(s*s') - invBy*s' - s*invBy')
end
```

[41]:
```
InvBFGSUpdate (generic function with 1 method)
```

1.2 Illustration

[35]:
```
using LinearAlgebra

n = 2
B = invB = zeros(n,n)+I
n, m = size(B)
n
```

[35]:
```
2
```

[36]:
```
s = [-1.75,-0.75]
y = [-8.5,-5.0]
```

[36]:
```
2-element Array{Float64,1}:
 -8.5
 -5.0
```

[37]:
```
Bp = BFGSUpdate(B, y, s)
```

[37]:
```
2×2 Array{Float64,2}:
 4.03437  1.91981
 1.91981  2.18711
```

[38]:
```
inv(Bp)
```

[38]:
```
2×2 Array{Float64,2}:
  0.425679  -0.373654
 -0.373654   0.785212
```

[39]:
```
invBp = InvBFGSUpdate_naive(invB, y, s)
```

[39]:
```
2×2 Array{Float64,2}:
  0.425679  -0.373654
 -0.373654   0.785212
```

[42]:
```
invBp = InvBFGSUpdate(invB, y, s)
```

[42]:
```
2×2 Array{Float64,2}:
  0.425679  -0.373654
 -0.373654   0.785212
```

[43]:
```
using BenchmarkTools
```

[44]:
```
@btime InvBFGSUpdate_naive(invB, y, s)
```

```
  508.813 ns (10 allocations: 928 bytes)
```

[44]:
```
2×2 Array{Float64,2}:
  0.425679  -0.373654
 -0.373654   0.785212
```

[45]:
```
@btime InvBFGSUpdate(invB, y, s)
```

```
  621.104 ns (12 allocations: 1.02 KiB)
```

[45]:
```
2×2 Array{Float64,2}:
  0.425679  -0.373654
 -0.373654   0.785212
```

[48]:
```
n = 1000
B = invB = zeros(n,n)+I
n, m = size(B)
s = rand(n)
y = 10 * rand(n)

A1 = InvBFGSUpdate_naive(invB, y, s)

@btime InvBFGSUpdate_naive(invB, y, s)
```

```
  130.074 ms (18 allocations: 61.04 MiB)
```

[48]:
```
1000×1000 Array{Float64,2}:
  0.998174      0.00019233   -0.0024911    …  -0.00178234   -0.0013405
  0.00019233    1.00313      -0.00298919       0.0010108     0.00199688
 -0.0024911    -0.00298919    0.999956        -0.00328072   -0.00374312
 -0.000538096   0.00253257   -0.00328819       0.000121238   0.00106254
 -0.0020016     0.00042971   -0.00295659      -0.00189669   -0.00134064
 -0.000341807  -0.000242411  -0.000179128  …  -0.000406352  -0.000414848
 -0.00214846   -0.00246343   -0.000156532     -0.00279954   -0.0031608
 -0.000732262  -0.000794224  -0.000100173     -0.000942319  -0.00105058
  0.000249249   0.0033472    -0.0031399        0.00112417    0.00216899
 -0.0022609    -0.00243981   -0.000322084     -0.00290623   -0.00323643
 -0.000518353   0.00140116   -0.00209627   …  -0.000154378   0.000412183
 -0.00164088   -0.0014498    -0.000564819     -0.00202544   -0.00215994
 -0.00156023   -0.000180419  -0.001773        -0.00161304   -0.00134843

 -0.00205167   -0.00148629   -0.00104299      -0.00244725   -0.00250848
 -0.00196983    0.0007554    -0.00325267      -0.00177977   -0.00112361
 -0.00232968   -0.00238106   -0.000469058  …  -0.00295992   -0.00325659
 -0.0020054     0.000569904  -0.00310598      -0.0018639    -0.00126113
 -0.000304224   0.000837788  -0.00124624      -8.65733e-5    0.000251003
 -1.38881e-5    0.00155878   -0.00162543       0.000393085   0.000906619
 -0.001178      0.00103745   -0.00254936      -0.000911403  -0.000327132
 -0.00193834   -0.000382604  -0.00203921   …  -0.00204532   -0.0017685
 -0.00121552    0.00165616   -0.00323472      -0.000787504   7.24088e-6
 -0.0007674     0.00200454   -0.00303141      -0.000246781   0.000569118
 -0.00178234    0.0010108    -0.00328072       0.998475     -0.000823974
 -0.0013405     0.00199688   -0.00374312      -0.000823974   1.00011
```

[49]:
```
A2 = InvBFGSUpdate(invB, y, s)

@btime InvBFGSUpdate(invB, y, s)
```

```
  54.159 ms (20 allocations: 61.04 MiB)
```

[49]:
```
1000×1000 Array{Float64,2}:
  0.998174      0.00019233   -0.0024911    …  -0.00178234   -0.0013405
  0.00019233    1.00313      -0.00298919       0.0010108     0.00199688
 -0.0024911    -0.00298919    0.999956        -0.00328072   -0.00374312
 -0.000538096   0.00253257   -0.00328819       0.000121238   0.00106254
 -0.0020016     0.00042971   -0.00295659      -0.00189669   -0.00134064
 -0.000341807  -0.000242411  -0.000179128  …  -0.000406352  -0.000414848
 -0.00214846   -0.00246343   -0.000156532     -0.00279954   -0.0031608
 -0.000732262  -0.000794224  -0.000100173     -0.000942319  -0.00105058
  0.000249249   0.0033472    -0.0031399        0.00112417    0.00216899
 -0.0022609    -0.00243981   -0.000322084     -0.00290623   -0.00323643
 -0.000518353   0.00140116   -0.00209627   …  -0.000154378   0.000412183
 -0.00164088   -0.0014498    -0.000564819     -0.00202544   -0.00215994
 -0.00156023   -0.000180419  -0.001773        -0.00161304   -0.00134843

 -0.00205167   -0.00148629   -0.00104299      -0.00244725   -0.00250848
 -0.00196983    0.0007554    -0.00325267      -0.00177977   -0.00112361
 -0.00232968   -0.00238106   -0.000469058  …  -0.00295992   -0.00325659
 -0.0020054     0.000569904  -0.00310598      -0.0018639    -0.00126113
 -0.000304224   0.000837788  -0.00124624      -8.65733e-5    0.000251003
 -1.38881e-5    0.00155878   -0.00162543       0.000393085   0.000906619
 -0.001178      0.00103745   -0.00254936      -0.000911403  -0.000327132
 -0.00193834   -0.000382604  -0.00203921   …  -0.00204532   -0.0017685
 -0.00121552    0.00165616   -0.00323472      -0.000787504   7.24088e-6
 -0.0007674     0.00200454   -0.00303141      -0.000246781   0.000569118
 -0.00178234    0.0010108    -0.00328072       0.998475     -0.000823974
 -0.0013405     0.00199688   -0.00374312      -0.000823974   1.00011
```

[50]:
```
A1-A2
```

[50]:
```
1000×1000 Array{Float64,2}:
  4.44089e-16  -9.21572e-19  -2.1684e-18   …   1.30104e-18  -6.50521e-19
 -8.40257e-19   8.88178e-16  -8.67362e-19      -1.0842e-18    4.33681e-19
 -1.73472e-18  -8.67362e-19  -3.33067e-16      -8.67362e-19  -4.33681e-19
  3.25261e-19   8.67362e-19   4.33681e-19      -1.21973e-19   4.33681e-19
 -8.67362e-19   1.30104e-18  -1.73472e-18       2.1684e-19   -8.67362e-19
  5.42101e-19   1.6263e-19   -5.42101e-20   …  -2.71051e-19  -1.6263e-19
  0.0           4.33681e-19   0.0               1.30104e-18   1.30104e-18
 -2.1684e-19   -4.33681e-19   5.42101e-20      -1.0842e-19   -4.33681e-19
 -2.76472e-18   0.0          -8.67362e-19       8.67362e-19  -8.67362e-19
 -2.1684e-18   -8.67362e-19   5.42101e-20      -4.33681e-19   1.30104e-18
  0.0          -2.1684e-19   -2.1684e-18   …  -1.0842e-19    1.6263e-19
  2.1684e-19    1.73472e-18   3.25261e-19      -4.33681e-19   4.33681e-19
  4.33681e-19   1.35525e-18   6.50521e-19       4.33681e-19   2.1684e-19

  0.0          -6.50521e-19  -4.33681e-19       4.33681e-19  -4.33681e-19
  4.33681e-19   1.19262e-18   8.67362e-19      -4.33681e-19  -1.51788e-18
```

```
-8.67362e-19   0.0           0.0          …   4.33681e-19   0.0
-8.67362e-19   3.25261e-19   3.46945e-18      1.30104e-18  -1.30104e-18
-1.6263e-19   -4.33681e-19   4.33681e-19     -5.01444e-19   5.42101e-20
 8.14846e-19   0.0          -8.67362e-19      0.0           1.0842e-19
 2.1684e-18    2.1684e-19    4.33681e-19      1.0842e-19    5.42101e-19
-6.50521e-19  -1.0842e-19   -1.73472e-18  …   4.33681e-19  -8.67362e-19
 8.67362e-19  -2.1684e-19   -4.33681e-19     -3.25261e-19   2.74439e-19
 1.73472e-18   0.0           1.73472e-18      9.21572e-19  -2.1684e-19
 1.30104e-18  -6.50521e-19  -8.67362e-19      0.0          -6.50521e-19
-6.50521e-19   4.33681e-19  -4.33681e-19     -6.50521e-19   0.0
```

[51]:
```
norm(A1-A2)
```

[51]:
```
9.666376776492351e-15
```

1.3 SR1 Update

Adapted from https://en.wikipedia.org/wiki/Symmetric_rank-one

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k},$$

where $y_k = \nabla f(x_{\text{cand}}) - \nabla f(x_k)$ and $s_k = x_{\text{cand}} - x_k$.

Here B_k is not necessarily positive definite.

The corresponding update to the approximate inverse-Hessian $H_k = B_k^{-1}$ is

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}.$$

The SR1 formula has been rediscovered a number of times. A drawback is that the denominator can vanish. Some authors have suggested that the update be applied only if

$$|s_k^T (y_k - B_k s_k)| \geq r \|s_k\| \|y_k - B_k s_k\|,$$

where $r \in (0, 1)$ is a small number, e.g. 10^{-8}.