Lecture FYS4411, February 23, 2024

FYS4411/9411 Feb 23, 2024

Taylor expansion around $\hat{\alpha}$

$$
\begin{aligned}
\mathbb{E}[E_2(\hat{\alpha})] &= E(\hat{\alpha}) \\
&= E(\alpha_{k+1}) + \nabla E(\alpha_{k+1})^T \\
&\quad \times (\hat{\alpha}-\alpha_{k+1}) \\
&\quad + \frac{1}{2}(\hat{\alpha}-\alpha_{k+1})^T \nabla^2 E \\
&\quad \times (\hat{\alpha}-\alpha_{k+1}) \\
&\quad + O(\|\hat{\alpha}-\alpha_{k+1}\|^3)
\end{aligned}
$$

generalize $E(\hat{\alpha}) \rightarrow f(x)$

$$f(y) = f(x) + \nabla^T f(x)(y-x)$$

$$+ \frac{1}{2}(y-x)^T \nabla^2 f (y-x)$$

$$\nabla^2 f \geq m \mathbb{1}$$

in a more compact way

$$f(x) = c + b^T x + \frac{1}{2} x^T A x$$

$$\frac{\partial f}{\partial x} = 0 \Rightarrow Ax = -b \Rightarrow$$

$$x = A^{-1} b$$

Newton-Raphson's method

$$x_{k+1} = x_k - \left(\nabla f(x_k)\right)^{-1} f(x_k)$$

$$(\nabla f)_{ij} = \frac{\partial f_i}{\partial x_j} \quad \text{Jacobian.}$$

(our case $f(x) = \nabla E(\alpha)$)

Drawback:

- we need $\nabla f(x_k)$
  (in our case a 2nd derivative)
- sensitive to initial value of the iteration, x_0

First simplification:

Gradient descent

$$x_{k+1} = x_k - \gamma_k \underbrace{f(x_k)}_{\nabla E(\alpha)}$$

$$\alpha_{k+1} = \alpha_k - \gamma_k \nabla E(\alpha)$$

γ_k : learning rate.

if we can diagonalize A

$$\gamma_k < \frac{2}{\lambda_{max}(A)}$$

Broyden's method, secant method:

Newton's method transform $f(x)$ into a fixed point problem $x = g(x)$

$$g(x) = x - f(x)/f'(x)$$

and hence

$$g'(x) = f(x)\,f''(x)/[f'(x)]^2$$

if x^* is a simple root

then $g'(x^*) = 0$ if $f(x^*) = 0$
and $f'(x^*) \neq 0$

Newton's method requires
that we have to calculate
$f(x_k)$ and $f'(x_k)$

in the secant method
the derivative is approximated

$$f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

$$x_{k+1} = x_k - f(x_k)\left[\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}\right]$$

Broyden's algorithm

$$x_{k+1} = x_k - \underline{J(x_k)}^{-1} f(x_k)$$

has a cost of n^2 ($J \in \mathbb{R}^{n \times n}$) floating point operations + matrix inversion $O(n^3)$ floating point operations.

Normal not invert, but solve

$$J(x_k)\, s_k = -f(x_k)$$

$$\left(x_{k+1} = x_k - J^{-1} f(x_k) \right)$$

$$x_{k+1} = x_k + s_k$$

Algorithm:

Define x_0 and initial Jacobian B_0

For $k=0$, MaxIter

$$B_k s_k = -f(x_k)$$

$$x_{k+1} = x_k + s_k$$

$$y_k = f(x_{k+1}) - f(x_k)$$

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

End For

$$B_k s_k = -f(x_k)$$

$$B_{k+1} = B_k + \underbrace{C_k} \quad \Big| \quad C_k = u_k s_k^T$$

undetermined

using secant method

$$u_k = \frac{y_k - B_k s_k}{s_k^T s_k}$$

$$\left(\begin{aligned} s_k &= x_{k+1} - x_k \\ y_k &= f(x_{k+1}) - f(x_k) \end{aligned} \right)$$

$$(x_{k+1} - x_k)^T u = 0$$

$$B_{k+1} u = B_k \cdot u \quad \text{if}$$

Broyden-Fletcher-Goldfarb-Shanno
(BFGS)

update $\quad B_{k+1} = B_k + a u u^T$

$\quad\quad\quad\quad + b v v^T$

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

Steepest descent

$$f(x) = c + \frac{1}{2} x^T A x - x^T b$$

A known, b known

$$\frac{\partial f}{\partial x} = 0 \Rightarrow Ax = b$$

solve iteratively:

Define an initial x_0

Define the error

$$r_k = b - Ax_k$$

for $i=0$, Maxiter

$$x_{k+1} = \boxed{\alpha_k r_k + x_k} \; ?$$

$$r_{k+1} = b - Ax_{k+1}$$

end for

How do we find α_k

$$x_{k+1} = x_k + \alpha_k r_k$$

$$r_{k+1} = b - Ax_{k+1}$$
$$= b - A(x_k + \alpha_k r_k)$$

r_{k+1} is orthogonal to r_k

$$r_k^T r_{k+1} = 0 \Rightarrow$$

$$r_k^T r_k = r_k^T \alpha_k A r_k \Rightarrow$$

$$\boxed{\alpha_k = \frac{r_k^T r_k}{r_k^T A r_k}}$$

use to update x_{k+1}

$$x_{k+1} = x_k + \alpha_k r_k$$