Optimization Theory and Methods

Prof. Xiaoe Ruan

Email: wruanxe@xjtu.edu.cn
Tel:13279321898

2019, Fall

Chapter 5. Nonlinear Least-Square Method

Gauss-Newton Method

Quasi-Newton update of Gauss-Newton Matrix

Hybrid Algorithm

Decomposed Quasi-Newton Method

Curve fitting problem:

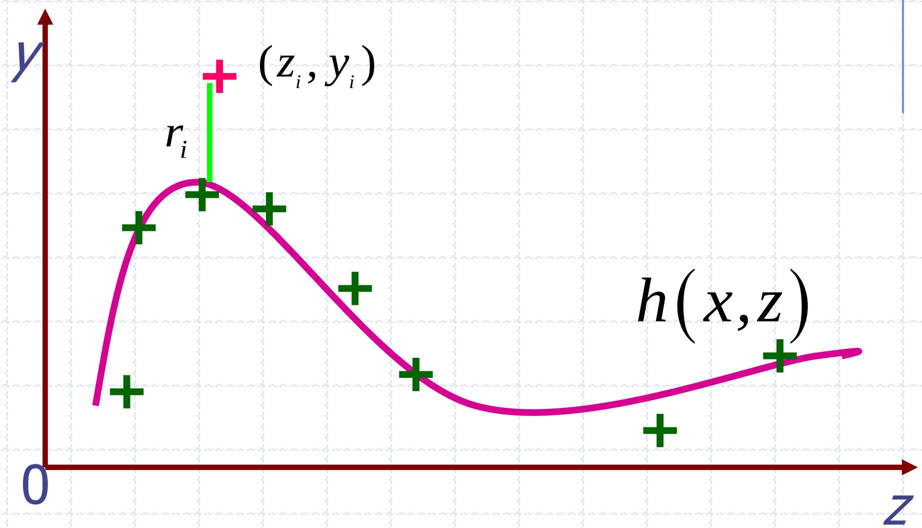
Given data $(z_i, y_i) \in R^{l+1}$ **where** $z_i \in R^l, y_i \in R.$

$z_i \neq z_j \ (i \neq j), i, j = 1, \cdots, m.$ **Find a nonlinear function**

$h(x, z)$ **so that**

it approximates all of given data under some criterion

----- optimal curve fitting.



Least-Square Criterion:

$$\min_{x \in D} f(x) = \sum_{i=1}^{m} \left(h(x, z_i) - y_i \right)^2 = \sum_{i=1}^{m} \left(r_i(x) \right)^2$$

$$r(x) = \left[r_1(x), r_2(x), \cdots, r_m(x) \right]^{\mathrm{T}}$$ **---residual vector.**

1、Gauss-Newton Algorithm

Nonlinear Least-Square Optimization $\min\limits_{x\in D} f(x)=\frac{1}{2}\sum_{i=1}^{m}\left(r_i(x)\right)^2$

Denote $r(x)=\left[r_1(x),r_2(x),\cdots,r_m(x)\right]^{\mathrm{T}}$, $A(x)=\left(\nabla r_1(x),\cdots,\nabla r_m(x)\right)$.

Then $g(x) \triangleq \nabla f(x)=\sum_{i=1}^{m}\nabla r_i(x) r_i(x) = A(x)r(x)$

$$G(x) \triangleq \nabla^2 f(x)=\underbrace{\boxed{A(x)A(x)^{\mathrm{T}}}}_{M(x)}+\underbrace{\boxed{\sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)}}_{S(x)}$$

If $f\left(x^*\right)=\min\limits_{x\in D} f(x)$, **then it is possible that** $r_i\left(x^*\right)=0.$

Thus, if $x\in N\left(x^*\right)$, **then** $G(x)\approx A(x)A(x)^{\mathrm{T}} = M(x).$

Let $q_k(d)=f\left(x^{(k)}+d\right) = f\left(x^{(k)}\right)+\nabla f\left(x^{(k)}\right)^{\mathrm{T}} d+\frac{1}{2}d^{\mathrm{T}}\nabla^2 f\left(\xi^{(k)}\right)d$

If $q_k\left(d^{(k)}\right) = \min q_k(d)$ **Then**

$$A\left(x^{(k)}\right)A\left(x^{(k)}\right)^{\mathrm{T}} d^{(k)} \approx \nabla^2 f\left(\xi^{(k)}\right)d^{(k)} = -\nabla f\left(x^{(k)}\right) = -A\left(x^{(k)}\right)r\left(x^{(k)}\right)$$

From $A\left(x^{(k)}\right)A\left(x^{(k)}\right)^{\mathrm{T}} d^{(k)} = -g^{(k)} = -A\left(x^{(k)}\right)r\left(x^{(k)}\right)$ **yields** $d^{(k)}$.

Gauss-Newton algorithm: $x^{(k+1)} = x^{(k)} + d^{(k)}$.

If $d^{(k)}$ **is descending direction of** $f(x)$,

damped G-N algorithm: $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$.

Features:

(1) $G(x) \approx A(x)A(x)^{\mathrm{T}} \triangleq M(x)$ **only 1-st-order derivative**

(2) $M_k \triangleq M\left(x^{(k)}\right) = A\left(x^{(k)}\right)A\left(x^{(k)}\right)^{\mathrm{T}} \triangleq A_k A_k^{\mathrm{T}}$ **semi-PD,**

i.e. $d^{(k)}$ **is possibly descending direction.**

(3) Compared with Newton's Method, convergence is influenced.

Th1. Assume that (1) $f(x)$ is 2-nd-order continuously differentiable on open convex set D

(2) There exists $x^* \in D$ s.t. $g(x^*) = A(x^*) r(x^*) = 0.$

(3) $A(x)$ and $G(x)$ satisfying Lipschitz continuous on D

i.e. $\exists \beta, \gamma$ s.t. $\|A(x) - A(y)\| \le \beta \|x - y\|, \forall x, y \in D,$

and $\|G(x) - G(y)\| \le \gamma \|x - y\|, \forall x, y \in D,$

(4) $A(x), x \in D$ is full rank and $\|A(x)\| < \sigma, \forall x \in D;$

$\|H(x)\| = \left\|M(x)^{-1}\right\| < M, \forall x \in D;$ where $\sigma, M = \text{const.}$

Then, G-N sequence $x^{(k+1)} = x^{(k)} + d^{(k)}$ is meaningful and

$$\left\|x^{(k+1)} - x^*\right\| \le \left\|H(x^*) S(x^*)\right\| \left\|x^{(k)} - x^*\right\| + O\left(\left\|x^{(k)} - x^*\right\|^2\right)$$

Proof: From $A(x), x \in D$ full rank implies $M(x) = A(x)A(x)^{\mathrm{T}}$ is invertible. Then $d^{(k)}$ is available.

i.e. $x^{(k+1)} = x^{(k)} + d^{(k)}$ is meaningful.

Further $\|A(x) - A(y)\| \le \beta \|x - y\|, \forall x, y \in D,$

$$\|G(x) - G(y)\| \le \gamma \|x - y\|, \forall x, y \in D, \quad \|A(x)\| < \sigma, \forall x \in D;$$

Then $\|M(x) - M(y)\|$

$$= \left\|A(x)A(x)^{\mathrm{T}} - A(x)A(y)^{\mathrm{T}} + A(x)A(y)^{\mathrm{T}} - A(y)A(y)^{\mathrm{T}}\right\| \le 2\sigma\beta \|x - y\|,$$

Therefore $\|H(x) - H(y)\| = \left\|M(x)^{-1} - M(y)^{-1}\right\|$

$$= \left\|M(x)^{-1}\left[M(y) - M(x)\right]M(y)^{-1}\right\| \le 2M^2\sigma\beta \|x - y\|,$$

$$\|S(x) - S(y)\| = \left\|\left[G(x) - M(x)\right] - \left[G(y) - M(y)\right]\right\|$$

$$\le (\gamma + 2\sigma\beta)\|x - y\|, \quad \forall x, y \in D.$$

whilst $$0=g\left(x^{*}\right)=g\left(x^{(k)}\right)+G\left(x^{(k)}\right)\left(x^{*}-x^{(k)}\right)+O_{1}\left(\left\|x^{(k)}-x^{*}\right\|^{2}\right)$$

$$=g\left(x^{(k)}\right)+\left[M\left(x^{(k)}\right)+S\left(x^{(k)}\right)\right]\left(x^{*}-x^{(k)}\right)+O_{1}\left(\left\|x^{(k)}-x^{*}\right\|^{2}\right)$$

i.e. $$M\left(x^{(k)}\right)\left(x^{(k)}-x^{*}\right)=g\left(x^{(k)}\right)-S\left(x^{(k)}\right)\left(x^{(k)}-x^{*}\right)+O_{1}\left(\left\|x^{(k)}-x^{*}\right\|^{2}\right)$$

Therefore $$x^{(k+1)}-x^{*}-\left(x^{(k+1)}-x^{(k)}\right)=x^{(k)}-x^{*}$$

$$=M\left(x^{(k)}\right)^{-1}g\left(x^{(k)}\right)-M\left(x^{(k)}\right)^{-1}S\left(x^{(k)}\right)\left(x^{(k)}-x^{*}\right)+O_{2}\left(\left\|x^{(k)}-x^{*}\right\|^{2}\right)$$

Then $$x^{(k+1)}-x^{*}=\left(x^{(k+1)}-x^{(k)}\right)+M\left(x^{(k)}\right)^{-1}g\left(x^{(k)}\right)$$

$$-H\left(x^{(k)}\right)S\left(x^{(k)}\right)\left(x^{(k)}-x^{*}\right)+O_{2}\left(\left\|x^{(k)}-x^{*}\right\|^{2}\right)$$

Additionally $$x^{(k+1)}-x^{(k)}=d^{(k)}=-M\left(x^{(k)}\right)^{-1}g^{(k)}$$ **Then**

$$x^{(k+1)}-x^{*}=-H\left(x^{*}\right)S\left(x^{*}\right)\left(x^{(k)}-x^{*}\right)-H\left(x^{(k)}\right)\left[S\left(x^{(k)}\right)-S\left(x^{*}\right)\right]\left(x^{(k)}-x^{*}\right)$$

$$-\left[H\left(x^{(k)}\right)-H\left(x^{*}\right)\right]S\left(x^{*}\right)\left(x^{(k)}-x^{*}\right)+O_{2}\left(\left\|x^{(k)}-x^{*}\right\|^{2}\right)$$

Thus $$\left\|x^{(k+1)}-x^{*}\right\|\leq\left\|H\left(x^{*}\right)S\left(x^{*}\right)\right\|\left\|x^{(k)}-x^{*}\right\|+O\left(\left\|x^{(k)}-x^{*}\right\|^{2}\right)$$

Discussion: $\left\|x^{(k+1)}-x^*\right\| \le \left\|H\left(x^*\right)S\left(x^*\right)\right\|\left\|x^{(k)}-x^*\right\|+O\left(\left\|x^{(k)}-x^*\right\|^2\right)$

Let $\rho=\left\|H\left(x^*\right)S\left(x^*\right)\right\|,\quad O\left(\left\|x^{(k)}-x^*\right\|^2\right)\le c\left\|x^{(k)}-x^*\right\|\left\|x^{(k)}-x^*\right\|$

Then $\left\|x^{(k+1)}-x^*\right\| \le \left(\rho+c\left\|x^{(k)}-x^*\right\|\right)\left\|x^{(k)}-x^*\right\|.$

Thus (1) **If** $A(x), x\in D$ **is full rank and** $g\left(x^*\right)=A\left(x^*\right)r\left(x^*\right)=0.$

Then $r\left(x^*\right)=0.$ **i.e.** $S\left(x^*\right)=\sum_{i=1}^{m} r_i\left(x^*\right)\nabla^2 r_i\left(x^*\right)=0$ **i.e.** $\rho=0.$

Then the (damped) G-N algorithm is quadratically convergent.

(2) **If** $A(x), x\in D$ **is not full rank but** $g\left(x^*\right)=A\left(x^*\right)r\left(x^*\right)=0.$

then $r\left(x^*\right)\ne 0.$ **i.e.** $\rho\ne 0.$ **If** $\rho<1,\quad \left\|x^{(k)}-x^*\right\| \ll 1,$

s.t. $\rho+c\left\|x^{(k)}-x^*\right\|<1$ **then G-N algorithm is linearly convergent.**

(3) **If** $\rho=\left\|H\left(x^*\right)S\left(x^*\right)\right\|\ge 1$ **i.e.** $\left\|S\left(x^*\right)\right\|$ **is larger**

Then Gauss-Newton algorithm is divergent.

2、Quasi-Newton update of G-N Matrix

$$G(x) \triangleq \nabla^2 f(x) = \underbrace{A(x)A(x)^{\mathrm{T}}}_{M(x)} + \underbrace{\sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)}_{S(x)}$$

Assume $f(x^*) = \min_x f(x)$,

$$g(x^*) = A(x^*)r(x^*) = 0.$$

If $A(x^*)$ **singular, then** $r(x^*) \neq 0.$

Thus

$$G(x^{(k)}) = M(x^{(k)}) + S(x^{(k)}) \approx \underbrace{M(x^{(k)})}_{M_k} + \underbrace{\sum_{i=1}^{m} r_i(x^{(k)}) W_i(x^{(k)})}_{C_k}$$

where $W_i(x^{(k)}) \approx \nabla^2 r_i(x^{(k)})$ **updated by SR1 formula**

Note $q_k(d) = f(x^{(k)} + d) = f(x^{(k)}) + \nabla f(x^{(k)})^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} \nabla^2 f(\xi^{(k)}) d$

Let $\nabla^2 f(\xi^{(k)}) = M_k + C_k$. **From** $q_k(d^{(k)}) = \min q_k(d)$ **induces** $d^{(k)}$.

-----Brown-Dennis Gauss-Newton matrix SR1 update.

Conclusion:

Brown-Dennis Gauss-Newton SR1 algorithm is convergent.

If $r\left(x^{*}\right)=0,$ **B-D G-N algorithm is quadratically convergent.**

If the memory is enough, the method is realizable.

Idea: **Determine** $W_i\left(x^{(k)}\right)=\nabla^2 r_i\left(x^{(k)}\right)$ **by update formula**

Requirement : **Quasi-N Eq.** $B_{k+1}\delta^{(k)}=y^{(k)}.$ $\delta^{(k)}=x^{(k+1)}-x^{(k)}$

As $$y^{(k)} \triangleq \nabla f\left(x^{(k+1)}\right)-\nabla f\left(x^{(k)}\right)=\nabla^2 f\left(\bar{x}^{(k)}\right)\left(x^{(k+1)}-x^{(k)}\right)$$

$$=G\left(\bar{x}^{(k)}\right)\left(x^{(k+1)}-x^{(k)}\right)=\left[M\left(\bar{x}^{(k)}\right)+S\left(\bar{x}^{(k)}\right)\right]\left(x^{(k+1)}-x^{(k)}\right)$$

Thus $S\left(\bar{x}^{(k)}\right)\left(x^{(k+1)}-x^{(k)}\right)=y^{(k)}-M\left(\bar{x}^{(k)}\right)\left(x^{(k+1)}-x^{(k)}\right).$ **i.e.**

quasi-N Eq. $C_{k+1}\delta^{(k)}=\gamma^{(k)}$ **is satisfied.** $\gamma^{(k)}=y^{(k)}-M\left(\bar{x}^{(k)}\right)\left(x^{(k+1)}-x^{(k)}\right).$

Difference of $M_k,\gamma^{(k)}$ **produce different updates.**

Generalization 1:

Brown-Dennis Gauss-Newton SR2 update:

Let $M\left(\bar{x}^{(k)}\right)=A\left(x^{(k+1)}\right)A\left(x^{(k+1)}\right)^{\mathrm{T}} \triangleq M_k$

$$\gamma^{(k)}=g^{(k+1)}-g^{(k)}-M_k\delta^{(k)}.$$

$$S\left(\bar{x}^{(k)}\right)=C_{k+1}=C_k+\Delta C_k$$ updated by SR2 formula

Generalization 2:

Betts Gauss-Newton update:

$$M_k=A\left(x^{(k)}\right)A\left(x^{(k)}\right)^{\mathrm{T}},\qquad \gamma^{(k)}=g^{(k+1)}-g^{(k)}-M_k\delta^{(k)}.$$

$$S\left(\bar{x}^{(k)}\right)=C_{k+1}=C_k+\Delta C_k$$ updated by quasi-N SR2 formula

Gen.3: **Bartholomew-Biggs, Dennis-Gay-Welsch Gauss-Newton update**

$$M_k = A\left(x^{(k+1)}\right)A\left(x^{(k+1)}\right)^{\mathrm{T}}, \ \gamma^{(k)} = \left[A_{k+1} - A_k\right]r^{(k+1)}. \ \ \eta^{(k)} = \gamma^{(k)} - C_k\delta^{(k)},$$

$$S\left(\bar{x}^{(k)}\right) = C_{k+1} \ = C_k + \frac{y^{(k)}\eta^{(k)\mathrm{T}} + \eta^{(k)}y^{(k)\mathrm{T}}}{\delta^{(k)\mathrm{T}}y^{(k)}} - \frac{\delta^{(k)\mathrm{T}}\eta^{(k)}}{\left[\delta^{(k)\mathrm{T}}y^{(k)}\right]^2}y^{(k)}y^{(k)\mathrm{T}}$$

Gen.4: **Gill-Murray Gauss-Newton update**

$$M_k = A\left(x^{(k+1)}\right)A\left(x^{(k+1)}\right)^{\mathrm{T}}, \ \gamma^{(k)} = g^{(k+1)} - g^{(k)} - M_k\delta^{(k)}, \ W_k = A_{k+1}A_{k+1}^{\mathrm{T}} + C_k,$$

$$S\left(\bar{x}^{(k)}\right) = C_{k+1} \ = C_k + \frac{y^{(k)}y^{(k)\mathrm{T}}}{\delta^{(k)\mathrm{T}}y^{(k)}} - \frac{W_k\delta^{(k)}\delta^{(k)\mathrm{T}}W_k}{\delta^{(k)\mathrm{T}}W_k\delta^{(k)}}$$

Conclusion:

Gill-Murray Gauss-Newton update-based sequence is locally superlinearly convergent.

Remarks:

(1) $S\left(\overline{x}^{(k)}\right)=C_{k+1}$ **may be inappropriate, Efficiency ??**

May employ tuning factor such as $C_{k+1} = \tau_k C_k + \Delta C_k$

(2) C_k **May be non-PD. Thus** $G_k = M_k + C_k$ **may be non-PD.**

Hence $d^{(k)}$ **may be non-descending direction.**

Possible ways:

Trust-region technique or decomposed quasi-Newton method.

(3) At the beginning iterations, adopt Gauss-Newton method.

Then switch to Gauss-Newton quasi-N update-based method.

(4) Solve Eq. of searching direction by proper manner.

3. Hybrid Algorithm

Combine Gauss-Newton algorithm with some unconstrained optimization method and automatically switch to a proper method.------Switching method.

Powell's hybrid algorithm

Rotate Gauss-Newton direction to negative gradient for searching direction.

Gauss-Newton direction:

Let $q_k(d) = f\left(x^{(k)}+d\right) = f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} \nabla^2 f\left(\xi^{(k)}\right) d$

If $q_k\left(d^{(k)}\right) = \min q_k(d)$ **Then**

$A\left(x^{(k)}\right) A\left(x^{(k)}\right)^{\mathrm{T}} d^{(k)} = -A\left(x^{(k)}\right) r\left(x^{(k)}\right)$ **achieving** $d^{(k)}$.

Powell's Hybrid Algorithm

solve $d_{GN}^{(k)}$ **from** $A_k A_k^T d^{(k)} = -A_k r^{(k)}$

$$\left\| d_{GN}^{(k)} \right\| \le \Delta_k$$

Yes

$$d^{(k)} = d_{GN}^{(k)}$$

No

From $\min \sum_{i=1}^{m} \left[r\left(x^{(k)}\right) - \alpha \nabla r\left(x^{(k)}\right)^{\mathrm{T}} \frac{g^{(k)}}{\left\| g^{(k)} \right\|} \right]^2$

get $\alpha_k = \dfrac{\left\| g^{(k)} \right\|^3}{g^{(k)\mathrm{T}} A_k A_k^{\mathrm{T}} g^{(k)}}$

$$\alpha_k \ge \Delta_k$$

Yes

$$d^{(k)} = -\Delta_k \frac{g^{(k)}}{\left\| g^{(k)} \right\|}$$

No

$$d_{CP}^{(k)} = -\alpha_k \frac{g^{(k)}}{\left\| g^{(k)} \right\|}$$

From $\left\| d_{CP}^{(k)} + \lambda \left(d_{GN}^{(k)} - d_{CP}^{(k)} \right) \right\| = \Delta_k$ **get** λ_k

$$d^{(k)} = d_{CP}^{(k)} + \lambda_k \left(d_{GN}^{(k)} - d_{CP}^{(k)} \right)$$

Construction of $d^{(k)}$ for $\alpha_k < \Delta_k$

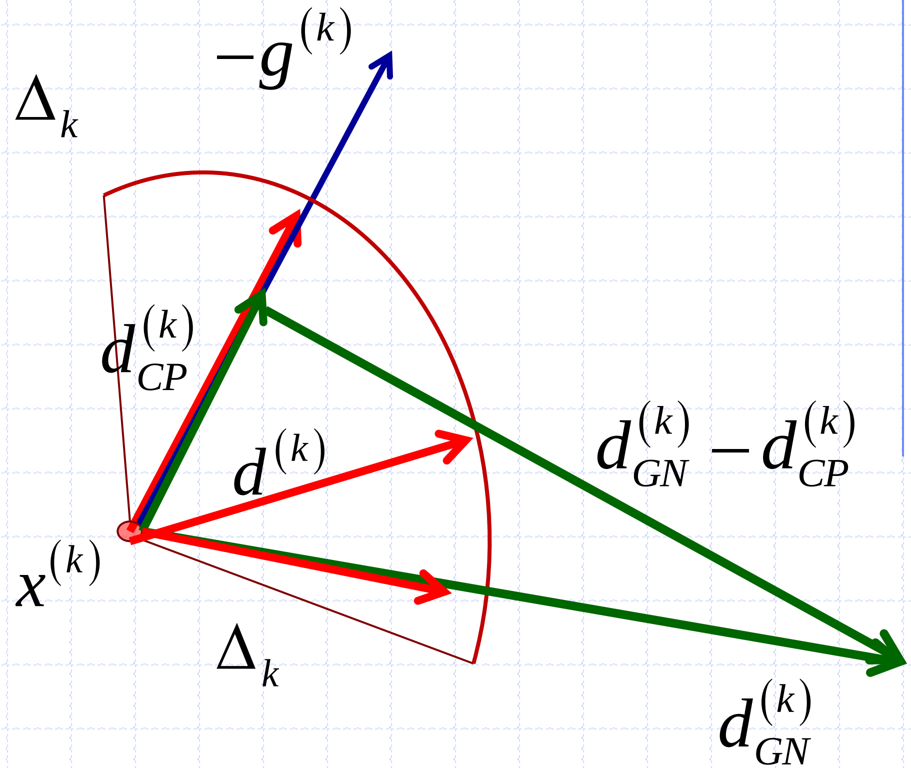
Then

If $f\left(x^{(k)} + d^{(k)}\right) < f\left(x^{(k)}\right)$

Let $x^{(k+1)} = x^{(k)} + d^{(k)}.$

Otherwise,

tuning $\Delta_k.$



Powell proved that the hybrid algorithm is convergent but only modification of Gauss-Newton method with no much improvement of convergence rate.

Comparison:

Gauss-Newton Algorithm	BFGS quasi-N Algorithm
Utilizing 1-st-order derivative to estimate 2-nd derivative according to structure of LS problem.	Utilizing 1-st-order derivative to estimate 2-nd derivative of objective fcn but needs multi-update.
Quadratic convergent for zero-residual pbm.	Superlinear convergent for zero-residual pbm.
Linearly convergent for smaller-residual pbm and faster near minimizer.	Superlinearly convergent for smaller-residual pbm and slower near minimizer.
Not or linearly convergent for larger-residual pbm.	Superlinearly convergent for larger-residual pbm.

Al-Baali-Fletcher Hybrid Algorithm

Combine Gauss-Newton M with BFGS quasi-N M for searching direction.

BFGS update $$B_k = BFGS\left(B_{k-1}, \delta^{(k-1)}, y^{(k-1)}\right)$$

$$= B_{k-1} + \frac{y^{(k-1)} y^{(k-1)\mathrm{T}}}{y^{(k-1)\mathrm{T}} \delta^{(k-1)}} - \frac{B_{k-1} \delta^{(k-1)} \delta^{(k-1)\mathrm{T}} B_{k-1}^{\mathrm{T}}}{\delta^{(k-1)\mathrm{T}} B_{k-1} \delta^{(k-1)}}$$

Switching criterion of Al-Baali-Fletcher hybrid algorithm:

Let $$\Delta(B, \delta, y) = \left(\frac{y^{\mathrm{T}} B^{-1} y}{\delta^{\mathrm{T}} y}\right)^2 - \frac{2\delta^{\mathrm{T}} y}{\delta^{\mathrm{T}} B \delta} + 1$$

----approximation degree of B **to** $\nabla^2 f(x)$

Switching criterion: $\Delta\left(B_{k-1}, \delta^{(k-1)}, y^{(k-1)}\right) < \Delta\left(M_k, \delta^{(k-1)}, y^{(k-1)}\right)$

Let $q_k(d) = f\left(x^{(k)}+d\right) = f\left(x^{(k)}\right)+\nabla f\left(x^{(k)}\right)^{\mathrm{T}} d+\frac{1}{2}d^{\mathrm{T}}B_k d,$

From $q_k\left(d^{(k)}\right)=\min q_k(d)$ **yields** $d^{(k)}=-B_k^{-1}g^{(k)}.$

Al-Baali-Fletcher Hybrid Algorithm

$$\Delta\left(B_{k-1},\delta^{(k-1)},y^{(k-1)}\right) < \Delta\left(M_k,\delta^{(k-1)},y^{(k-1)}\right)$$

No

$$B_k = M_k$$

Yes

$$B_k = BFGS\left(B_{k-1},\delta^{(k-1)},y^{(k-1)}\right)$$

$BFGS\left(B_{k-1},\delta^{(k-1)},y^{(k-1)}\right)$ **better than** M_k

$$d^{(k)}=-B_k^{-1}g^{(k)}$$

$$x^{(k+1)}=x^{(k)}+\alpha_k d^{(k)}$$

Remarks:

(1)ABF hybrid algorithm guarantees B_k PD. Thus

$d^{(k)}$ is descending direction. α_k determined by simple method.

(2) Al-Baali-Fletcher hybrid algorithm is effective.

(3) For smaller-residual pbm, G-N algorithm is quadratically convergent, whilst for larger-residual pbm, BFGS algorithm is superlinearly convergent. Therefore,

ABF hybrid algorithm does not guarantee superlinear convergence.

Fletcher-Xu switching criterion: $$\frac{f\left(x^{(k-1)}\right)-f\left(x^{(k)}\right)}{f\left(x^{(k-1)}\right)}<\rho, \quad \rho\in(0,1).$$

Recommendation: $\rho=0.2$. Then

ABF hybrid algorithm eventually switches to BFGS algorithm which means Flecher-Xu algorithm is superlinearly convergent.

4、Decomposed Quasi-Newton Algorithm

$$G(x) \triangleq \nabla^2 f(x) = \underbrace{A(x)A(x)^{\mathrm{T}}}_{M(x)} + \underbrace{\sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)}_{S(x)}$$

$$G\left(x^{(k)}\right) = M\left(x^{(k)}\right) + S\left(x^{(k)}\right) = M\left(x^{(k)}\right) + \sum_{i=1}^{m} r_i\left(x^{(k)}\right)\nabla^2 r_i\left(x^{(k)}\right)$$

$$M\left(x^{(k)}\right) = A_k A_k^{\mathrm{T}}, \qquad \sum_{i=1}^{m} r_i\left(x^{(k)}\right)\nabla^2 r_i\left(x^{(k)}\right) \approx L_k L_k^{\mathrm{T}} + L_k A_k^{\mathrm{T}} + A_k L_k^{\mathrm{T}}$$

Let $q_k(d) = f\left(x^{(k)} + d\right)$

$$= f\left(x^{(k)}\right) + \nabla f\left(x^{(k)}\right)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} B_k d,$$

$$B_k = A_k A_k^{\mathrm{T}} + L_k L_k^{\mathrm{T}} + L_k A_k^{\mathrm{T}} + A_k L_k^{\mathrm{T}} = \left(A_k + L_k\right)\left(A_k + L_k\right)^{\mathrm{T}}.$$

From $q_k\left(d^{(k)}\right) = \min q_k(d)$ **reaches** $d^{(k)}$ **s.t.** $B_k d^{(k)} = -A_k^{\mathrm{T}} r^{(k)}$

Decomposed quasi-Newton algorithm: $x^{(k+1)} = x^{(k)} + d^{(k)}.$

Discussion:

(1) If row rank of $A_k + L_k$ is full, then $B_k = (A_k + L_k)(A_k + L_k)^{\mathrm{T}}$ PD.

Thus $d^{(k)}$ is descending direction.

(2) Quasi-N Eq. $B_k \delta^{(k)} = (A_k + L_k)(A_k + L_k)^{\mathrm{T}} \delta^{(k)} = y^{(k)}.$

(3) If $\delta^{(k)\mathrm{T}} y^{(k)} \neq 0,$ then $B_k \delta^{(k)} = y^{(k)}$ has solution.

Hence

$$(A_k + L_k)(A_k + L_k)^{\mathrm{T}} \delta^{(k)} = y^{(k)} \Longrightarrow \begin{cases} (A_k + L_k)h = y^{(k)} \\ (A_k + L_k)^{\mathrm{T}} \delta^{(k)} = h \\ h^{\mathrm{T}} h = \delta^{(k)\mathrm{T}} y^{(k)} \end{cases}$$

Update L_{k+1} by construction of h

Denote $R_{k+1} = A_{k+1} + L_{k+1}$, $V_k = A_{k+1} + L_k$, $W_k = V_k V_k^{\mathrm{T}}$.

If W_k **nonsingular.** **Let** $h = aV_k^{\mathrm{T}}\delta^{(k)} + bV_k^{\mathrm{T}}W_k^{-1}y^{(k)}$,

From $\left(A_{k+1} + L_{k+1}\right)h = y^{(k)}$ **yields** $A_{k+1} + L_{k+1} = X_0 + \bar{Y}$.

$$X_0 = a\frac{y^{(k)}\delta^{(k)\mathrm{T}}V_k}{\delta^{(k)\mathrm{T}}y^{(k)}} + b\frac{y^{(k)}y^{(k)\mathrm{T}}W_k^{-1}V_k}{\delta^{(k)\mathrm{T}}y^{(k)}},$$

$$\bar{Y} = cV_k\left[I - \frac{V_k^{\mathrm{T}}\delta^{(k)}\delta^{(k)\mathrm{T}}V_k}{\delta^{(k)\mathrm{T}}W_k\delta^{(k)}}\right] + d\left[I - \frac{y^{(k)}\delta^{(k)\mathrm{T}}}{\delta^{(k)\mathrm{T}}y^{(k)}}\right]V_k.$$

where a,b,c,d **to be determined satisfying**

$$a^2\delta^{(k)\mathrm{T}}W_k\delta^{(k)} + 2ab\delta^{(k)\mathrm{T}}y^{(k)} + b^2y^{(k)\mathrm{T}}W_k^{-1}y^{(k)} = \delta^{(k)\mathrm{T}}y^{(k)},$$

$$ad\delta^{(k)\mathrm{T}}W_k\delta^{(k)} - bc\delta^{(k)\mathrm{T}}y^{(k)} = 0, \qquad c+d=1.$$

Update of L_{k+1}

$$L_{k+1} = L_k + (a-d)\frac{y^{(k)}\delta^{(k)\mathrm{T}}V_k}{\delta^{(k)\mathrm{T}}y^{(k)}} + b\frac{y^{(k)}y^{(k)\mathrm{T}}W_k^{-1}V_k}{\delta^{(k)\mathrm{T}}y^{(k)}} - c\frac{W_k\delta^{(k)}\delta^{(k)\mathrm{T}}V_k}{\delta^{(k)\mathrm{T}}W_k\delta^{(k)}}$$

Substituting into $B_{k+1} = (A_{k+1} + L_{k+1})(A_{k+1} + L_{k+1})^{\mathrm{T}}$ **achieves** B_{k+1}

Difference of update formula depends on choices of a, b, c, d

Feature: **Update invariance under linear transformation.**

Conclusion: **For** $f(x) = \frac{1}{2}\sum_{i=1}^{m} r_i^2(x)$,

when $A(x), G(x)$ **satisfy some conditions**

Decomposed quasi-Newton algorithm is linearly convergent.

THANK YOU FOR ATTENDING