Mathematics in Quasi-Newton Method

Peng Du and Xijun Wang

Department of Computing and Software
McMaster University

April 2, 2003

Outline

- Elementary Quasi-Newton Method
- Update for Cholesky Factorization
- Update for Limited Memory
- Update for Sparse Hessians
- Update for Reduced Hessians
- References

Elementary Quasi-Newton Method

Unconstrained Optimization: $\min f(x)$

Quasi-Newton Iteration: $x^{k+1} = x^k - t_k H^k g^k, H^{k+1} = H^k + E^k$

Notation: $B, B^*, H, H^*, x, x^*, g, g^*, \sigma(p) = x^* - x, y = g^* - g$

Desired Update Properties:

- Quasi-Newton Property: $B^*\sigma = y(H^*y = \sigma)$
- Symmetric
- Positive Definite

SR1: $B^* = B + \frac{\gamma\gamma^T}{\gamma^T\sigma}, \gamma = y - B\sigma$

DFP: $B^* = B - \frac{B\sigma y^T + y\sigma^T B}{\sigma^T y} + \left(1 + \frac{\sigma^T B\sigma}{\sigma^T y}\right)\frac{yy^T}{\sigma^T y}$

BFGS: $B^* = B + \frac{yy^T}{y^T\sigma} - \frac{B\sigma\sigma^T B}{\sigma^T B\sigma}$

PSB: $B^* = B + \frac{\gamma\sigma^T + \sigma\gamma^T}{\sigma^T\sigma} - \frac{\gamma^T\sigma}{\sigma^T\sigma}\frac{\sigma\sigma^T}{\sigma^T\sigma}$

Cholesky Factor Modification I — Rank-one

Cholesky Factor Modification (symmetric rank one P.D.) update

Using Classical Cholesky Factorization

$A = LDL^T \longrightarrow \bar{A} = A + \alpha z z^T \stackrel{?}{=} \bar{L}\bar{D}\bar{L}^T$

$\bar{A} = L(D + \alpha p p^T)L^T, Lp = z$

Factorize $D + \alpha p p^T = \tilde{L}\tilde{D}\tilde{L}^T$ giving $\bar{A} = \bar{L}\bar{D}\bar{L}^T, \bar{L} = L\tilde{L}, \bar{D} = \tilde{D}$

$\tilde{L}$ is of the special form

$$L(p, \beta, \gamma) = \begin{pmatrix} \gamma_1 & & & & & \\ p_2\beta_1 & \gamma_2 & & & & \\ p_3\beta_1 & p_3\beta_2 & \gamma_3 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ & & & & \gamma_{n-1} & \\ p_n\beta_1 & p_n\beta_2 & p_n\beta_3 & \vdots & p_n\beta_{n-1} & \gamma_n \end{pmatrix}$$

Assertion 1: Need only to compute $\tilde{d}_i, \beta_i$, in $O(n)$ time.

Assertion 2: Product of L and $\tilde{L}$ can be done in $O(n^2)$ time.

Drawback instable

Cholesky Factor Modification II — Rank-one

Using Givens Matrices (I)

2-dimensional case

$$\begin{pmatrix} c & s \\ s & -c \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} +\rho \\ 0 \end{pmatrix}$$

where $\rho^2 = z_1^2 + z_2^2, \rho = sign(z_1)(\rho^2)^{1/2}, c = z_1/\rho, s = z_2/\rho$.

n-dimensional case

$$P_j^i z = \begin{pmatrix} & & & (i)\text{th} & & & & (j)\text{th} & & & \\ 1 & & & & & & & & & & \\ & \ddots & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & c & & & & s & & & \\ & & & & 1 & & & & & & \\ & & & & & \ddots & & & & & \\ & & & & & & 1 & & & & \\ & & & s & & & & -c & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ \vdots \\ z_i \\ \vdots \\ z_j \\ \vdots \\ z_n \end{pmatrix} = \begin{pmatrix} z_1 \\ \vdots \\ \tilde{z}_i \\ \vdots \\ 0 \\ \vdots \\ z_n \end{pmatrix}$$

Sequential reduction

$$P_2^1 P_3^2 \cdots P_n^{n-1} z = \rho e_1 \text{ or } P_2^1 P_3^1 \cdots P_n^1 z = \rho e_1$$

Products of $P_n^{n-1} P_{n-1}^{n-2} \cdots P_3^2 P_2^1$ is of the special form

$$H_L(p,\beta,\gamma) = \begin{pmatrix} p_1\beta_1 & \gamma_1 & & & & \\ p_2\beta_1 & p_2\beta_2 & \gamma_2 & & & \\ p_3\beta_1 & p_3\beta_2 & p_3\beta_3 & \ddots & & \\ \vdots & \vdots & \vdots & & \gamma_{n-2} & \\ p_{n-1}\beta_1 & p_{n-1}\beta_2 & p_{n-1}\beta_3 & \cdots & p_{n-1}\beta_{n-1} & \gamma_{n-1} \\ p_n\beta_1 & p_n\beta_2 & p_n\beta_3 & \cdots & p_n\beta_{n-1} & p_n\beta_n \end{pmatrix}$$

Denote $H_U(\beta,p,\gamma) \triangleq H_L^T(p,\beta,\gamma)$

Factorization $A = R^T R \longrightarrow \bar{A} = A + \alpha z z^T \stackrel{?}{=} \bar{R}^T \bar{R}$

$$\bar{A} = R^T(I + \alpha p p^T)R, R^T p = z$$

Choose a sequence of Givens matrices such that

$$Pp = P_2^1 P_3^2 \cdots P_n^{n-1} p = \rho e_1$$

$\bar{A}$ can be written as

$$\bar{A} = R^T P^T P(I + \alpha p p^T) P^T P R = R^T P^T (I + \alpha \rho^2 e_1 e_1^T) P R = H^T J^T J H$$

where $H = PR$, J is identity matrix except $J_{11} = (1 + \alpha p^T p)^{1/2}$.

Assertion Product of P and R can be done in $O(n^2)$ time.

Let $\bar{H} = JH$, then $\bar{A} = \bar{H}^T \bar{H}$.

Choose a second sequence of Givens matrices such that

$$\bar{P}\bar{H} = \bar{P}_n^{n-1} \bar{P}_{n-1}^{n-2} \cdots \bar{P}_3^2 \bar{P}_2^1 \bar{H} = \bar{R}$$

Then $\bar{A} = \bar{R}^T \bar{R}$.

Cholesky Factor Modification III — Rank-one

Using Givens Matrices (II) $A = R^T R \longrightarrow \bar{A} = A + \alpha z z^T \stackrel{?}{=} \bar{R}^T \bar{R}$

$$\bar{A} = R^T(I + \alpha p p^T)R = R^T(I + \sigma p p^T)(I + \sigma p p^T)R$$

where $R^T p = z$ and $\sigma = \alpha/(1 + (1 + \alpha p^T p)^{1/2})$.

Choose a sequence of Givens matrices such that

$$P p = P_2^1 P_3^2 \cdots P_n^{n-1} p = \rho e_1$$

$\bar{A}$ can be written as

$$\bar{A} = R^T(I + \sigma p p^T)P^T P(I + \sigma p p^T)R = R^T H^T H R$$

where $H = P(I + \sigma p p^T) = P + \sigma \rho e_1 p^T$.

Note that $P = H_U(\beta, p, \gamma)$ for some β, γ. Then

$$H = H_U(\beta, p, \gamma) + \sigma \rho e_1 p^T = H_U(\bar{\beta}, p, \gamma)$$

where $\bar{\beta}$ differs from β only in the first element: $\bar{\beta} = \beta + \sigma \rho e_1$.

Choose a second sequence of Givens matrices such that

$$\bar{P} H = \bar{P}_n^{n-1} \bar{P}_{n-1}^{n-2} \cdots \bar{P}_3^2 \bar{P}_2^1 H = \tilde{R}$$

Assertion 1: $\tilde{R}$ is of the form $\tilde{R} = R(\tilde{\beta}, p, \tilde{\gamma}) \triangleq L(p, \tilde{\beta}, \tilde{\gamma})$

Then $\bar{A} = R^T H^T \bar{P}^T \bar{P} H R = R^T \tilde{R}^T \tilde{R} R$.

Assertion 2: Product of $\tilde{R}$ and R can be done in $O(n^2)$ time.

Cholesky Factor Modification IV — Rank-two

Product Form of BFGS

$B^+ = (I + vu^T)B(I + uv^T), u = p + \alpha B^{-1}y, v = \theta_1 Bp + \theta_2 y$
for some $\alpha, \theta_1, \theta_2$.

Triangularization of $I + zw^T$: $(I + zw^T)Q = \tilde{L}$

First choose a sequence of Givens matrices such that

$$Pw = P_2^1 P_3^2 \cdots P_n^{n-1} w = \rho e_1$$

Then $\bar{H} = (I + zw^T)P^T = P^T + \rho z e_1^T$ is lower Hessenberg matrix.

Choose a second sequence of Givens matrix to reduce the superdiagonal elements of $\bar{H}$ such that

$$\bar{H}\bar{P} = \tilde{L}$$

Assertion: $\tilde{L}$ is of the special form

$$\tilde{L}(w,\beta,z,\gamma,\lambda) = \begin{pmatrix} \lambda_1 & & & & \\ \beta_1 w_2 + \gamma_1 z_2 & \lambda_2 & & & \\ \beta_1 w_3 + \gamma_1 z_3 & \beta_2 w_3 + \gamma_2 z_3 & \ddots & & \\ \vdots & \vdots & & \lambda_{n-1} & \\ \beta_1 w_n + \gamma_1 z_n & \beta_2 w_n + \gamma_2 z_n & \cdots & \beta_{n-1} w_n + \gamma_{n-1} z_n & \lambda_n \end{pmatrix}$$

Factorization $B = L_1 D L_1^T \longrightarrow B^+ = (I + vu^T)B(I + uv^T) \stackrel{?}{=} \bar{L}_1 \bar{D} \bar{L}_1{}^T$

$B^+ = L(I + zw^T)(I + wz^T)L^T$, $L = L_1 D^{1/2}, Lz = v, Lw = LL^T u$.

Triangularize $I + zw^T = \tilde{L}Q^T$ giving

$$B^+ = L\tilde{L}Q^T Q\tilde{L}^T L^T = L^+ L^{+T}, L^+ = L\tilde{L} = L_1 D^{1/2}\tilde{L}$$

Assertion: $D^{1/2}\tilde{L}(w,\beta,z,\gamma,\lambda) = \tilde{L}(\hat{w},\beta,\hat{z},\gamma,\hat{\lambda}) = \tilde{L}(\hat{w},\hat{\beta},\hat{z},\hat{\gamma},e)\hat{D}^{1/2}$ where

$\hat{w} = D^{1/2}w, \hat{z} = D^{1/2}z, \hat{\lambda} = D^{1/2}\lambda, e = (1,\cdots,1)^T$,

$\hat{D}^{1/2} = diag(\hat{\lambda}_1,\cdots,\hat{\lambda}_n), \hat{\beta} = D^{-1/2}\beta, \hat{\gamma} = D^{-1/2}\gamma$.

Assertion: Product of $L_1\tilde{L}(\hat{w},\hat{\beta},\hat{z},\hat{\gamma},e)$ could be done in $O(n^2)$ time.

Cholesky Factor Modification V — Rank-two

BFGS $B^{+} = B + \sigma uu^T + \tau vv^T$

Using Classical Factorization

$$B = L_1 D L_1^T \longrightarrow B^{+} = B + \sigma uu^T + \tau vv^T \stackrel{?}{=} \bar{L}_1 \bar{D} \bar{L}_1^T$$

$$B^{+} = B + \sigma uu^T + \tau vv^T = L_1 (D + \sigma ww^T + \tau zz^T) L_1^T$$

where $L_1 w = u, L_1 z = v$.

Factorize $D + \sigma ww^T + \tau zz^T = \tilde{L}_1 \tilde{D} \tilde{L}_1^T$

Similar to the rank-one case, $\tilde{L}_1$ is of a special form.

Finally

$$L_1^{+} = L_1 \tilde{L}(w, \beta, z, \gamma, e), D^{+} = \tilde{D}$$

Assertion: Product of $L_1 \tilde{L}(w, \beta, z, \gamma, e)$ could be done in $O(n^2)$ time.

Implementing Quasi-Newton Method

Approach-1: Cholesky	Approach-2: Inverse Hessian
$B_k s_k = -\nabla f(x_k)$ $\Downarrow$ $R_k^T R_k = B_k$ $\Downarrow$ back and forward substitution $\Downarrow$ find R_{k+1} s.t. $R_{k+1}^T R_{k+1} = B_k + \Delta B_k$	$B_k s_k = -\nabla f(x_k)$ $\Downarrow$ $H_k = B_k^{-1}$ $\Downarrow$ solve for s_k is straight forward $\Downarrow$ update H_k directly, i.e. $H_{k+1} = f(H_k, \sigma_k, y_k) = B_{k+1}^{-1}$

Update Inverse Hessian Approximation: Properties Desired

Hessian Approximate	Inverse Hessian Approximate
$B_{k+1}\sigma_k = y_k$	$H_{k+1}y_k = \sigma_k$
B_{k+1} is positive definite	H_{k+1} is positive definite
B_{k+1} is symmetric	H_{k+1} is symmetric

- Requirements are in dual format. As a consequence,
- Update formula should be in dual format too

Updating Hessian Approximate

Broyden Family on B_k update

$$B_{k+1} = B_k - \frac{B_k\sigma_k(B_k\sigma_k)^T}{\sigma^T B_k\sigma_k} + \frac{y_k y_k^T}{y_k^T\sigma_k} + \theta_k(\sigma_k^T B_k\sigma_k)v_k v_k^T;$$

where $\theta_k \in [0,1]$ and

$$v_k = \left(\frac{y_k}{y_k^T\sigma_k} - \frac{B_k\sigma_k}{\sigma^T B_k\sigma^k}\right)$$

Broyden Family on H_k update

$$H_{k+1} = H_k - \frac{H_k y_k(H_k y_k)^T}{y^T H_k y_k} + \frac{\sigma_k\sigma_k^T}{\sigma_k^T y_k} + \phi_k(y_k^T H_k y_k)w_k w_k^T;$$

where $\phi_k \in [0,1]$ and

$$w_k = \left(\frac{\sigma_k}{\sigma_k^T y_k} - \frac{H_k y_k}{y^T H_k y^k}\right)$$

BFGS update of H_k when $\phi_k = 1$

$$H_{k+1} = H_k - \frac{H_k y_k (H_k y_k)^T}{y^T H_k y_k} + \frac{\sigma_k \sigma_k^T}{\sigma_k^T y_k} + (y_k^T H_k y_k) w_k w_k^T;$$

rewritten as:

$$H_{k+1} = V_k^T H_k V_k + \rho_k \sigma_k \sigma_k^T$$

where $\rho_k = \frac{1}{y_k^T \sigma_k}$, and $V_k = I - \rho y_k \sigma_k^T$.

Difficulties:

- H_k is dense usually
- H_k can be ill-conditioned

Solving Memory Problem – L-BFGS

$$H_{k+1} = V_k^T H_k V_k + \rho_k \sigma_k \sigma_k^T$$

expand $H_k = V_{k-1}^T H_{k-1} V_{k-1} + \rho_{k-1}\sigma_{k-1}\sigma_{k-1}^T$

$$\begin{aligned} H_{k+1} &= (V_k^T V_{k-1}^T) H_{k-1} (V_{k-1} V_k) \\ &\quad + \rho_{k-1} V_{k-1}^T \sigma_{k-1} \sigma_{k-1}^T V_{k-1} + \rho \sigma_k \sigma_k^T \end{aligned}$$

expand m times:

$$\begin{aligned} H_{k+1} &= (V_k^T \cdots V_{k-m}^T) H_{k-m} (V_{k-m} \cdots V_k) \\ &\quad + \rho_{k-m} (V_k^T \cdots V_{k-m+1}^T) \sigma_{k-m} \sigma_{k-m}^T (V_{k-m+1} \cdots V_k) \\ &\quad + \rho_{k-m+1} (V_k^T \cdots V_{k-m+2}^T) \sigma_{k-m+1} \sigma_{k-m+1}^T (V_{k-m+2} \cdots V_k) \\ &\quad \vdots \\ &\quad + \rho_k \sigma_k \sigma_k^T . \end{aligned}$$

Ideas of L-BFGS

- Maintain m most recent pairs of σ and y,
  $P = \{(\sigma_k, y_k), (\sigma_{k-1}, y_{k-1}), \ldots, (\sigma_{k-m+1}, y_{k-m+1})\}$

- At k-th iteration:

  If $(k \leq m)$
  $\quad P = P \cup \{(\sigma_k, y_k)\}$
  else
  $\quad P = (P - \{(\sigma_{k-m+1}, y_{k-m+1})\}) \cup \{(\sigma_k, y_k)\}$
  end

Performance of L-BFGS

P	N	L-BFGS m=5	L-BFGS m=7	L-BFGS m=9
1	1000	45/55 147/27/174	44/54 179/27/206	44/54 215/27/242
2	1000	53/58 165/337/502	55/58 237/394/631	57/59 288/381/669
4	100	106/111 35/3/38	94/98 42/5/47	57/61 27/2/29
5	100	134/168 43/14/57	126/147 55/10/65	111/131 51/17/68
7	50	162/164 25/50/75	148/150 35/40/75	150/152 39/41/80
10	961	168/280 516/630 1146	167/274 669/606 1275	163/267 680/610 1290
11	1000	36/42 116/37/153	35/41 139/35/174	34/40 162/35/197
12	100	254/260 93/145/238	245/251 112/146/258	246/252 133/149/282

the two numbers above: iterations/function-evaluations

there numbers below: iteration-time/function-time/total-time

Choosing Appropriate m

P	N	L-BFGS m=15	m=25	m=40
4	100	46/50 33/3/36	41/45 36/2/38	41/45 43/2/45
5	100	110/124 86/9/95	109/115 137/7/144	96/104 167/5/172
7	50	127/129 51/37/88	133/135 82/37/119	122/124 107/34/141
10	121	43/49 33/16/49	42/48 36/16/52	41/47 41/14/55
11	100	31/37 21/2/23	30/36 22/4/26	30/36 24/4/28
12	100	263/269 222/161/383	235/241 301/135/436	220/226 420/126/546

the two numbers above: iterations/function-evaluations

there numbers below: iteration-time/function-time/total-time

Dealing With H_k

- Choose of H_k influences the overall performance.

- Options found in literature:
  - Identity matrix $H_k = H_0 = I$
  - Identity matrix scaled at first step only $H_k = \frac{y_0^T \sigma_0}{\|y_0\|^2} I$
  - Identity matrix scaled at each step $H_k = \frac{y_k^T \sigma_k}{\|y_k\|^2} I$
  - Diagonal matrix minimize

$$H_k = argmin_{D \in diag} \|DY_{k-1} - S_{k-1}\|_F,$$

where $Y_{k-1} = [y_{k-1}, \ldots, y_{k-m}], S_{k-1} = [\sigma_{k-1}, \ldots, \sigma_{k-m}]$

Scaling Schemes Comparisons

P	N	M1	M2	M3	M4
1	1000	34/72 111/35/146	45/55 147/27/174	26/35 87/18/105	29/39 114/20/134
2	1000	51/54 165/330/495	53/58 165/337/502	48/50 160/329/489	50/55 175/332/507
7	50	89/179 14/52/66	162/164 25/50/75	111/119 18/34/52	119/121 25/35/60
10	961	214/569 674/1318/1992	168/280 516/630/1146	190/197 592/435/1027	174/179 544/405/949
11	1000	35/83 112/71/183	36/42 116/37/153	15/22 45/18/63	16/22 54/20/74
12	100	233/482 78/286/364	254/260 93/145/238	308/322 110/183/293	263/270 109/151/260
16	403	41/41 61/1205/1266	26/26 36/806/842	24/27 35/825/860	26/26 38/808/846

the two numbers above: iterations/function-evaluations

there numbers below: iteration-time/function-time/total-time

Update for Spare Hessian I

Suppose $\nabla^2 f(x)$ **has a known sparse pattern** K**:**

$(i,j) \in K \rightarrow [\nabla^2 f(x)]_{ij} = 0$

$(i,j) \in K \leftrightarrow (j,i) \in K, (\text{for } i,j = 1,2,\cdots,n)$

$(i,i) \notin K, (\text{for } i = 1,2,\cdots,n)$

Additional desired update property: $(i,j) \in K \rightarrow B^*_{ij} = 0$

$N = \{X \in \mathbb{R}^{n\times n} | Xp = y, X = X^T\}$

$S = \{X \in \mathbb{R}^{n\times n} | (i,j) \in K \rightarrow X_{ij} = 0\}$

$V = S \cap N$

Row Decomposition $B^* = B + \sum_{i=1}^n e_i e_i^T \triangle B$

$$B^* = B + \sum_{i=1}^n e_i e_i^T \left(\frac{yy^T}{p^T y} - \frac{Bpp^T B}{p^T Bp} \right) = B + \sum_{i=1}^n e_i \left(\frac{e_i^T y}{p^T y} y^T - \frac{e_i^T Bp}{p^T Bp} p^T B \right)$$

Update for Spare Hessian II

Symmetrization

$$\hat{B} = B + \frac{(y - Bp)z^T}{z^T p}$$

$$B^* = B + \frac{(y - Bp)z^T + z(y - Bp)^T}{z^T p} - \theta z z^T$$

where θ is chosen to assure QN condition:

$$\theta = \frac{(y - Bp)^T p}{(z^T p)^2}$$

$z = y\sqrt{p^T Bp/p^T y} + Bp \Longrightarrow$ BFGS

$z = y \Longrightarrow$ DFP

$z = y - Bp \Longrightarrow$ SR1

$z = p \Longrightarrow$ PSB

Update for Spare Hessian III

Update with desired sparse pattern

$$B^* = B + \sum_{i=1}^{n} e_i e_i^T \frac{(y - Bp) z_i^T}{z_i^T p}, z_i = D_i z$$

where D_i is the diagonal matrix with the jth component 1 if $B_{ij} \neq 0$, 0 if $B_{ij} = 0$.

Symmetrization

$$B^* = B + \sum_{i=1}^{n} \alpha_i (e_i z_i^T + z_i e_i^T)$$

where α_i is chosen to assure QN condition, satisfying

$$S\alpha = y - Bp$$

where $S = \sum_{i=1}^{n} z_i^T p e_i e_i^T + e_i^T p z_i e_i^T$.

When $z = y\sqrt{p^T Bp / p^T y} + Bp, y, y - Bp, p$, we get a sparse version of BFGS,DFP,SR1,PSB.

Drawback of symmetrization method

Update for Spare Hessian IV

Frobenius Norm Projection $\|X\|_F^2 \triangleq \sum_{i,j=1}^n X_{ij}^2 = Tr(X^T X)$

Define $\hat{B}^* = B^* + E$, and consider problem

$$\begin{array}{ll} \min & \frac{1}{2}\|E\|_F^2 = \frac{1}{2}Tr(E^T E) \\ s.t. & Ep = 0 \\ & E_{ij} = -B_{ij}^*, (i,j) \in K \\ & E = E^T \end{array}$$

We get

$$E = \frac{1}{2}\left(\sum_{i=1}^{n} e_i e_i^T (\mu p_i^T + p \mu_i^T) - 2B_K^*\right)$$

where $B_K^* = B^*$ if $(i,j) \in K$, $B_K^* = 0$ otherwise, $p_i = D_i p, \mu_i = D_i \mu$, and μ is chosen to assure QN condition

Store $\hat{B}^*$ and $r = 2B_K^* p$. Compute whole B^*.

Drawback (actually of all general sparse update) Can't minimize an n-dimensional quadratic in at most n updates with exact line search.

Can't guaranteed $\hat{B}^*$ to be P.D.

Consider $K = \{(i,j) | i \neq j, i, j = 1, \cdots, n\}$. Then

$$B^*_{ii} = \frac{y(i)}{p(i)}, i = 1, \cdots, n$$

In general, it is not possible to derive a diagonal and always P.D. update.

Question What are the maximum sparsity requirements we may impose so that there always exists a P.D. QN update?

Update for Spare Hessian V

Irreducible(Primitive) Matrices: There is no symmetric permutation of the rows and columns of the considered spare pattern such that the permuted pattern is block diagonal with at least two blocks.

Define perturbation matrix E as

$$E_{ij} = \delta_{ij}\|p_i\|^2 - p_i(j)p_j(i)$$

For $(i,j) \notin K$ and $i < j$, define matrix R^{ij} as (P.S.D.)

$$R^{ij} = \begin{pmatrix} & \vdots & & \vdots & \\ \cdots & p^2(j) & \cdots & -p(i)p(j) & \cdots \\ & \vdots & & \vdots & \\ \cdots & -p(i)p(j) & \cdots & p^2(i) & \cdots \\ & \vdots & & \vdots & \end{pmatrix}$$

Then

$$E = \sum_{(i,j)\notin K, i<j} R^{ij}$$

Properties of perturbation matrix E Suppose the given sparse pattern is irreducible and $p(i) \neq 0, i = 1, \cdots, n$. Then E satisfies

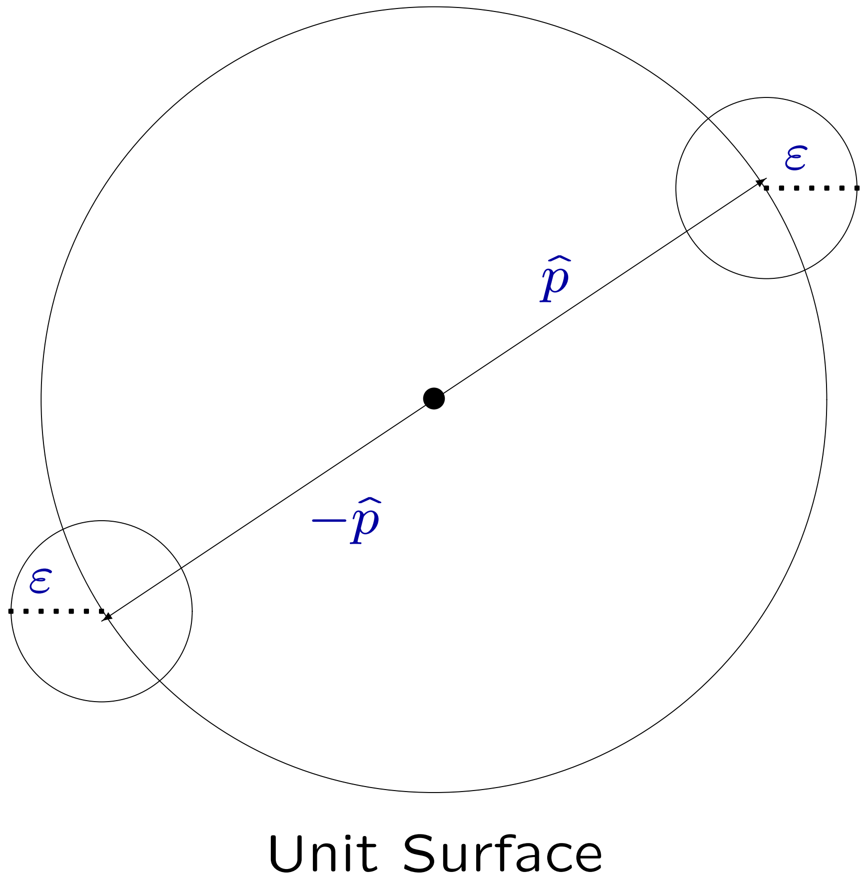
(i) E is symmetric.

(ii) E has the designated spare pattern.

(iii) $Ep = 0$.

(iv) For all $z \neq 0$ and $z \neq \lambda p$ where $\lambda \in \mathbb{R}, z^T E z > 0$.

Existence Theorem of P.D. Sparse Update Suppose the given sparse pattern is irreducible and $p(i) \neq 0, i = 1, \cdots, n$. If B+C is any sparse update, then there exists a $\bar{\tau} >= 0$ such that for all $\tau \geq \bar{\tau}$, matrix $B^* = B + C + \tau E$ is also a sparse update and P.D.



Note 1: $p(j) \neq 0$ for all j, sparse pattern is reducible.

Note 2: $p(j) = 0$ for some j, sparse pattern is irreducible.

I wish I am a Magician

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

$$\Downarrow$$

$$\Downarrow$$

$$\Downarrow$$

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

Magic Show – step (1)

Define:

- $\mathcal{G}_k = span\{g_0, g_1, \ldots, g_k\}$

- $\mathcal{G}_k^\perp$ denotes the orthogonal complement of $\mathcal{G}_k$ in R^n

Lemma:

Consider the Broyden method applied to a general nonlinear function. If $B_0 = \sigma I$ with $\sigma > 0$, then $s_k \in \mathcal{G}_k$ for all k. Moreover, if $z \in \mathcal{G}_k$ and $w \in \mathcal{G}_k^\perp$, then $B_k z \in \mathcal{G}_k$ and $B_k w = \sigma w$.

Magic Show – step (2)

Notation:

$r_k = dim(\mathcal{G}_k)$

$B_k \in R^{n \times r_k}$ forms a basis for $\mathcal{G}_k$

QR **factorization:**

$$B_k = Z_k T_k$$

where:

$Z_k \in R^{n \times r_k}$: an orthonormal basis matrix

$T_k \in R^{r_k \times r_k}$: a nonsingular upper triangular matrix

Magic Show – step (3)

More Notation:

$W_k \in R^{n\times(n-r_k)}$: an orthonormal basis matrix for $\mathcal{G}_k^{\perp}$

Define:

$$Q_k = (Z_k W_k)$$

A transformed Hessian Approximate:

$$Q_k^T B_k Q_k = \begin{pmatrix} Z_k^T B_k Z_k & (Z_k^T B_k W_k)^T \\ Z_k^T B_k W_k & W_k^T B_k W_k \end{pmatrix} = \begin{pmatrix} Z_k^T B_k Z_k & 0 \\ 0 & \sigma I_{n-r_k} \end{pmatrix}$$

A transformed gradient:

$$Q_k^T g_k = \begin{pmatrix} Z_k^T g_k \\ 0 \end{pmatrix}$$

Magic Show – step (4)

$$B_k s_k = -g_k$$

$$\Updownarrow$$

$$(Q_k^T B_k Q_k) Q_k^T s_k = -Q_k^T g_k$$

$$\Updownarrow$$

$$\begin{pmatrix} Z_k^T B_k Z_k & 0 \\ 0 & \sigma I_{n-r_k} \end{pmatrix} \begin{pmatrix} Z_k^T \\ W_k^T \end{pmatrix} \begin{pmatrix} q_k \\ q_k' \end{pmatrix} = - \begin{pmatrix} Z_k^T \\ 0 \end{pmatrix}$$

$$\Updownarrow$$

$$Z_k^T B_k Z_k q_k = -Z_k^T g_k, \quad s_k = Z_k q_k$$

Magic Show – step (5)

Fact: $Z_k^T B_k Z_k$ is positive definite if B_k is positive definite

Cholesky Factorization:

$$Z_k^T H_k Z_k = R_k^T R_k$$

$$\Downarrow$$

$$R_k^T R_k q_k = -Z_k^T g_k$$

Benefits:

- $Cond(Z_k^T H_k Z_k) \leq Cond(H_k)$
- Z_k, R_k require less memory
- Computing s_k requires less time

Update Z_k – the orthonormal basis

Step 1): Calculate $\rho_{k+1} = \|(I - Z_k Z_k^T) g_{k+1}\|$

Step 2): If $(\rho_{k+1} = 0)$

$$Z_{k+1} = Z_k, \quad r_{k+1} = r_k$$

else

$$z_{k+1} = \frac{(I - Z_k Z_k^T) g_{k+1}}{\rho_{k+1}}$$

$$Z_{k+1} = [Z_k \quad z_{k+1}]$$

Update R_k – the Cholesky factor

Want: $Z_{k+1}^T B_{k+1} Z_{k+1} = R_{k+1}^T R_{k+1}$

Two-Step Approach:

Expand Step: find R_k'' such that

$$(R_k'')^T R_k'' = Z_{k+1}^T B_k Z_{k+1}$$

Broyden-Update step: find R_{k+1} from R_k'' such that

$$Z_{k+1}^T B_{k+1} Z_{k+1} = R_{k+1}^T R_{k+1}$$

Expand step: $(R_k'')^T R_k'' = Z_{k+1} B_k Z_{k+1}$

We know $R_k : R_k^T R_k = Z_k^T B_k Z_k$.

$$Z_{K+1} B_k Z_{k+1} = \begin{pmatrix} Z_k^T B_k Z_k & Z_k^T B_k z_{k+1} \\ z_{k+1}^T B_k Z_k & \sigma \end{pmatrix}$$

$$= \begin{pmatrix} Z_k^T B_k Z_k & 0 \\ 0 & \sigma \end{pmatrix}$$

$$= \begin{pmatrix} R_k^T R_k & 0 \\ 0 & \sigma \end{pmatrix}$$

$$= \begin{pmatrix} R_k & 0 \\ 0 & \sigma^{1/2} \end{pmatrix}^T \begin{pmatrix} R_k & 0 \\ 0 & \sigma^{1/2} \end{pmatrix}$$

$$= (R_k'')^T R_k''$$

Expand Step – Continued

For Convenience, we define:

- $v_k = Z_k^T g_k$
- $u_k = Z_k^T g_{k+1}$
- $q_k = Z_k^T s_k$

Update them due to the expansion of Z_k

- $v_k'' = Z_{k+1}^T g_k = \begin{bmatrix} v_k \\ 0 \end{bmatrix}$
- $u_k'' = Z_{k+1}^T g_{k+1} = \begin{bmatrix} u_k \\ \rho_{k+1} \end{bmatrix}$
- $q_k'' = Z_{k+1}^T s_k = \begin{bmatrix} q_k \\ 0 \end{bmatrix}$

Broyden-Update Step

1. define: $$\sigma'_k = Z_{k+1}^T(x_{k+1} - x_k) = \alpha q''_k$$ $$y'_k = Z_{k+1}^T(g_{k+1} - g_k) = u''_k - v''_k$$

2. Apply QR factorization to
$$R''_k + w_1 w_2^T$$
where:
$$w_1 = \frac{1}{\|R''_k \sigma'_k\|} R''_k \sigma'_k$$
$$w_2 = \frac{1}{((y'_k)^T \sigma'_k)^{1/2}} y'_k - \frac{1}{\|R''_k \sigma'_k\|} (R''_k)^T R''_k \sigma'_k$$

Reduced Hessian Method (Complete Version)

Choose x_0 and $\sigma > 0$;

$k = 0; r_k = 1; g_k = \nabla(x_k)$;

$Z_k = \frac{g_k}{\|g_k\|}; R_k = \sigma^{1/2}; v_k = \|g_k\|$

while not converged **do**

Solve $R_k^T d_k = -v_k; R_k q_k = d_k$;

$s_k = Z_k q_k$;

Find α_k satisfying the Wolfe conditions;

$x_{k+1} = x_k + \alpha s_k; g_{k+1} = \nabla f(x_{k+1}); u_k = Z_k^T g_{k+1}$;

$(Z_{k+1}, r_{k+1}, R_k'', u_k'', v_k'', q_k'') = \mathit{expand}(Z_k, r_k, u_k, v_k, q_k, g_{k+1}, \sigma)$;

$(sigma_k' = \alpha_k q_k''; y_k' = u_k'' - v_k''$;

$R_{k+1} = \mathit{BroydenUpdate}(R_k'', \sigma_k', y_k')$;

$v_{k+1} = u_k''; k = k + 1$;

end do

References

- Gill, P. E., Golub, G. H., Murray, W. and Saunders, M. A. (1974). Methods for modifying matrix factorizations, *Mathematics of Computation* **28**(126): 505–535.

- Goldfarb, D. (1976). Factorized variable metric methods for unconstrained optimization, *Mathematics of Computation* **30**(136): 796–811.

- Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization, *Mathematical Programming* **45**: 503 –528.

- Shanno, D. F. (1980). On variable-metric methods for sparse Hessians, *Mathematics of Computation* **34**(150): 499–514.

- Toint, P. (1981). A note about sparsity exploiting Quasi-Newton, *Mathematical Programming* **21**: 172–181.

- Gill, P. E. and Leonard, M. W. (2001). Reduced-Hessian Quasi-Newton methods for unconstrained optimization, *SIAM J. OPTIM.* **12**(1): 209–237.