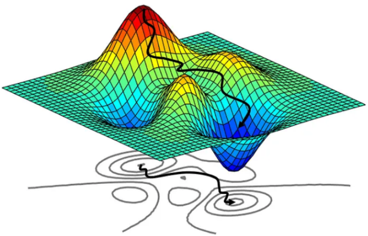
Almost multisecant BFGS quasi-Newton method

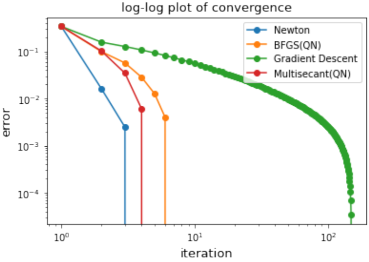
Mokhwa Lee, Yifan Sun

Stony Brook University

October 29, 2024

Motivation of Quasi-Newton (QN) method





Main Problem : $\min_{x \in \mathbb{R}^n} f(x)$ where f is differentiable

Method	Gradient Descent	Newton	Quasi-Newton(QN)
Convergence rate	linear, $O(C^n)$	quadratic, $O(C^{n^2})$	super-linear[1], $O(C^{n^{1.618}})$
Memory	O(n)	O(n^2)	O(n^2), $O(nL)$
Update x_{k+1}	$x_k - \alpha_k \nabla f(x_k)$	$x_k - \alpha_k [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$	$x_k - \alpha_k B_k^{-1} \nabla f(x_k)$
Algorithm	Efficient but slow	converges fast but expensive	L-BFGS, Broyden, etc

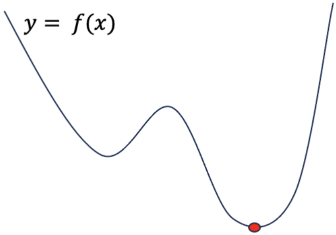
[1] Anton Rodomanov and Yurii Nesterov, Greedy Quasi-Newton Methods with Explicit Superlinear Convergence

Single Secant Condition

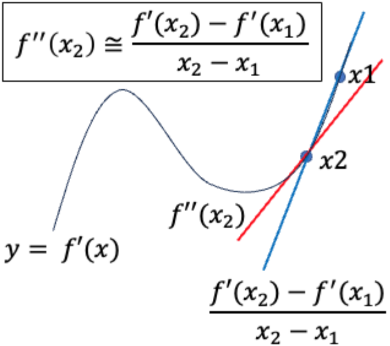
Second Order Taylor approximation for Hessian

$$\nabla^2 f(x_{k+1}) \approx B_{k+1} \quad : \quad \underbrace{B_{k+1}}_{\mathbb{R}^{n\times n}} (\underbrace{x_{k+1} - x_k}_{s_k \in \mathbb{R}^{n\times 1}}) = \underbrace{\nabla f(x_{k+1}) - \nabla f(x_k)}_{y_k \in \mathbb{R}^{n\times 1}}$$

Objective function graph



Gradient of objective function



Quasi-Newton update

Iterate update

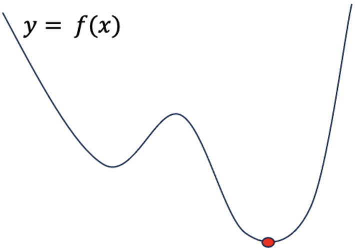
$$\begin{cases} x_{k+1} = x_k - \alpha B_k^{-1} \nabla f(x_k) \\ B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k) \qquad \text{(secant condition)} \end{cases}$$

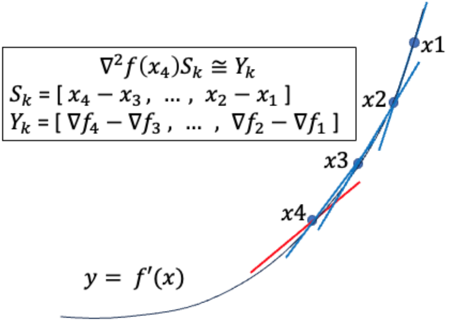
- Secant equation is under-determined.
- If B is symmetric, $\underbrace{\frac{n(n+1)}{2}}_{\#\text{ of vars}} > n$, we have $\frac{n(n-1)}{2}$ free variables.
- Secant equation has a unique solution in 1-dim since $\frac{1(1+1)}{2} = 1$.
- Several ways that satisfy secant condition by adding low-rank updates.

Multi-Secant Condition

Second Order Taylor approximation for Hessian

$$\underbrace{B_{k+1}}_{\mathbb{R}^{n\times n}} \underbrace{\left[\, s_k,\ s_{k-1},\ \ldots\ ,\ s_{k-p} \,\right]}_{S_k \in \mathbb{R}^{n\times p}} = \underbrace{\left[\, y_k,\ y_{k-1},\ \ldots\ ,\ y_{k-p} \,\right]}_{Y_k \in \mathbb{R}^{n\times p}}$$





- $S, Y \in \mathbb{R}^{n\times p}$ are low rank matrices, where p

Broyden–Fletcher–Goldfarb–Shanno algorithm(BFGS)

BFGS (single secant)

$$\mathrm{B}_{k+1} = B_k + \underbrace{\frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}}_{\text{low rank}}$$

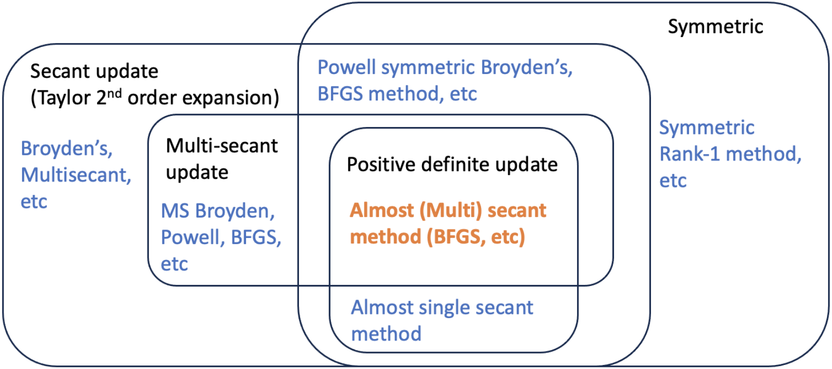
- Rank-2 update and satisfies secant condition.
- Maintain positive semidefiniteness of matrix B.

Approximate Hessian inverse(Sherman–Morrison–Woodbury formula)

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

- Iterate Update : $x_{k+1} = x_k - \alpha B_k^{-1} \nabla f(x_k)$
- Woodbury Matrix Inversion Lemma : $(A + UCV)^{-1} = A^{-1} - A^{-1}(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$

Quasi-Newton methods



- Use iterate and first order gradient information (no second order info)
- To maintain positive semidefinite hessian approximation, add $\mu > 0$
- Achieve stable and descent direction at each iteration (e.g. BFGS)
- Almost multisecant approximates the secant condition but maintain descent direction ($B_k \succeq 0 \ \Rightarrow\ -\nabla f_k^T B_k^{-1} \nabla f_k \leq 0$)

Quasi-Newton method comparison

Quasi-Newton : Update Hessian estimate

$$\mathrm{B}_{k+1} = B_k + \underbrace{f(B_k)}_{\textit{low rank}}$$

Method	Symmetric	PSD	Multisecant	$\mathrm{rank}(f(B_k))$
Broyden's	×	×	×	1
PSB[2]	✓	×	×	3
DFP[3]	✓	✓	×	3
BFGS[4]	✓	✓	×	2
Multisecant QN	×	×	✓	$2p$
Ours	✓	✓	$\approx$	$2p$

Table: Our method sacrifices the multisecant condition for PSD. The value p is a small number where $p \ll n$ and $2p$ is a low rank.

[2] PSB : Powell Symmetric Broyden
[3] DFP : Davidson Fletcher and Powell
[4] BFGS : Broyden, Fletcher, Goldfarb, Shannon

Multisecant BFGS

Multisecant BFGS : Hessian estimation

$$B_{k+1} = B_k + Y_k(Y_k^T S_k)^{-1} Y_k^T - B_k S_k (S_k^T B_k S_k)^{-1} S_k^T B_k$$

- Update $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$, where B_k^{-1} is derived from

$$\boxed{\begin{aligned} \arg\min_{B \in \mathbb{R}^{n\times n}} \quad & \|B - B_k\| \\ \text{s.t.} \quad & Bs = y \\ & B \succeq 0 \end{aligned}}$$

single secant($p = 1$)

$$\boxed{\begin{aligned} \arg\min_{B \in \mathbb{R}^{n\times n}} \quad & \|B - B_k\| \\ \text{s.t.} \quad & BS = Y \\ & B \succeq 0 \end{aligned}}$$

multi secant($p > 0$)

where

$$S = [x_{k+1} - x_k, \ \ldots, \ x_{k+1-p} - x_{k-p}] \in \mathbb{R}^{n\times p}$$

$$Y = [\nabla f(x_{k+1}) - \nabla f(x_k), \ \ldots, \ \nabla f(x_{k+1-p}) - \nabla f(x_{k-p})] \in \mathbb{R}^{n\times p}$$

- Only maintain the positive (semi)definiteness when $f(x)$ is quadratic.

Our Contribution : Almost Multisecant BFGS

Multisecant BFGS update

$$
\begin{aligned}
B_{k+1} &= B_k + Y_k(Y_k^T S_k)^{-1} Y_k^T - B_k S_k (S_k^T B_k S_k)^{-1} S_k^T B_k \\
&= B_k + (Y_k,\ B_k S_k) \begin{pmatrix} (Y_k^T S_k)^{-1} & 0 \\ 0 & -(S_k^T B_k S_k)^{-1} \end{pmatrix} \begin{pmatrix} Y_k^T \\ S_k^T B_k \end{pmatrix} \\
&= B_k - D_1 W^{-1} D_2^T
\end{aligned}
$$

• Multisecant QN does not guarantee symmetric positive semidefinite (PSD) Hessian estimate update.

• We symmetrize it and add μI to guarantee the positive semidefinite hessian estimate update (descent direction).

• $\bar{B}_{k+1} = \bar{B}_k \underbrace{- \frac{D_1 W^{-1} D_2^T + (D_1 W^{-1} D_2^T)^T}{2} + \mu I}_{\Delta \succeq 0} \in \mathbb{R}^{n \times n}$

• Find μ such that Δ is symmetric positive semidefinite (PSD)

Why did we choose BFGS for Multisecant extension?

$$B_{k+1} = B_k - D_1 W^{-1} D_2^T \quad \text{where} \quad Z_k = Y_k - B_k S_k$$

	D_1	D_2	W
Broyden's	Z_k	S_k	$-S_k^T S_k \in^{p\times p}$
PSB	$\begin{bmatrix} Z_k & S_k & S_k \end{bmatrix}$	$\begin{bmatrix} S_k & Z_k & S_k \end{bmatrix}$	$\begin{bmatrix} -S_k^T S_k & 0 & 0 \\ 0 & -S_k^T S_k & 0 \\ 0 & 0 & S_k^T S_k (Z_k^T S_k)^{-1} S_k^T S_k \end{bmatrix} \in^{3p\times 3p}$
DFP	$\begin{bmatrix} Z_k & Y_k & Y_k \end{bmatrix}$	$\begin{bmatrix} Y_k & Z_k & Y_k \end{bmatrix}$	$\begin{bmatrix} -Y_k^T S_k & 0 & 0 \\ 0 & -Y_k^T S_k & 0 \\ 0 & 0 & (Y_k^T S_k)(Z_k^T S_k)^{-1}(Y_k^T S_k) \end{bmatrix} \in^{3p\times 3p}$
BFGS	$\begin{bmatrix} Y_k & B_k S_k \end{bmatrix}$	$\begin{bmatrix} Y_k & B_k S_k \end{bmatrix}$	$\begin{bmatrix} -Y_k^T S_k & 0 \\ 0 & S_k^T B_k S_k \end{bmatrix} \in^{2p\times 2p}$
BFGS inv	$[H_k Y_k \quad S_k]$	$[H_k^T Y_k \quad S_k]$	$\begin{bmatrix} Y_k^T H_k Y_k + Y_k^T S & Y_k^T S_k \\ S_k^T Y_k & 0 \end{bmatrix} \in^{2p\times 2p}$

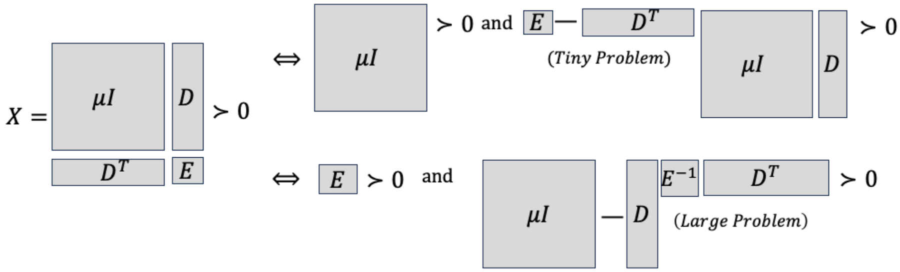
- $H_k = B_k^{-1}$
- W is not assumed to be symmetric nor PSD
- Challenging to apply Woodbury inversion lemma for other methods (Broyden, PSB, DFP).
- Woodbury approach is only possible for almost multisecant BFGS and BFGS inverse methods to compute W^{-1}.

Find μ by Schur Complement

By the Woodbury Inversion Lemma, we get almost multisecant BFGS

$$\bar{B}_{k+1}^{-1} = \bar{B}_k^{-1} - \frac{1}{2}\underbrace{\begin{bmatrix} D_1 & D_2\end{bmatrix}}_{D\in\mathbb{R}^{n\times 2p}} \underbrace{\begin{bmatrix} A_1 & W_k^{-1} \\ W_k^{-T} & A_2\end{bmatrix}}_{E\in\mathbb{R}^{2p\times 2p}} \underbrace{\begin{bmatrix} D_1^T \\ D_2^T\end{bmatrix}}_{D^T\in\mathbb{R}^{2p\times n}} + \mu I$$

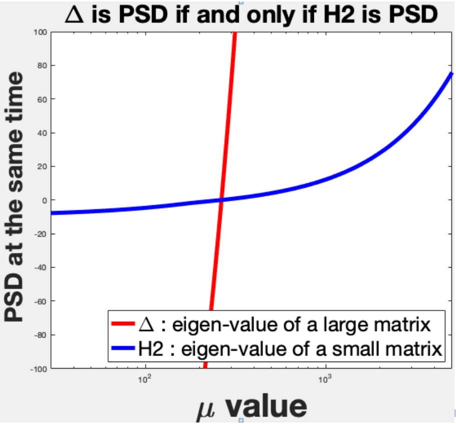
where $p \ll n$ (low rank update) for updating Hessian estimation.



- $\mu > 0$ satisfies $\Delta = \mu I - \frac{DED^T}{2} \succeq 0$ that ensures $B_{k+1}^{-1} \succeq 0$.

$$X = \begin{bmatrix} \mu I & D \\ D^T & E \end{bmatrix} \succ 0 \iff \mu I \succ 0 \text{ and } E - D^T \, \mu I \, D \succ 0 \quad (Tiny\ Problem)$$

$$\iff E \succ 0 \text{ and } \mu I - D \, E^{-1} \, D^T \succ 0 \quad (Large\ Problem)$$



Computation time of μ

m	n	p	AvgElapsedTime	StdElapsedTime
1000000	10000	5	0.082302468	0.012805515
1000000	10000	10	0.221266418	0.028299678
1000000	10000	15	0.342352725	0.045314651
1000000	100000	5	0.080595594	0.011917645
1000000	100000	10	0.262840806	0.067522336
1000000	100000	15	0.364708011	0.072498293
1000000	1000000	5	0.085991793	0.015292603
1000000	1000000	10	0.210551169	0.005418038
1000000	1000000	15	0.337621207	0.047064496
1000000	10000000	5	0.092226403	0.014109881
1000000	10000000	10	0.212651257	0.010707978
1000000	10000000	15	0.324698479	0.023220863
10000000	10000	5	1.342049018	0.198843912
10000000	10000	10	6.636063824	3.20097481
10000000	10000	15	27.95639044	11.13167948
10000000	100000	5	1.301018696	0.043209811
10000000	100000	10	3.268083497	0.705042952
10000000	100000	15	22.02416865	3.679317822
10000000	1000000	5	1.272271718	0.017935229
10000000	1000000	10	3.171529688	0.669478347
10000000	1000000	15	20.27326716	3.104544556
10000000	10000000	5	1.266372267	0.036094488
10000000	10000000	10	3.305674246	0.609465359
10000000	10000000	15	23.30601301	6.178500094

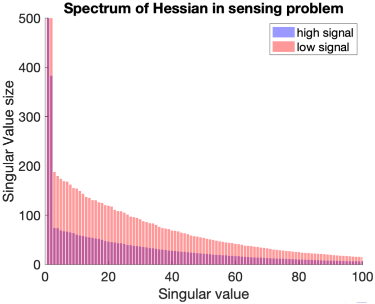
Table: Performance metrics for various values of m (number of data), n (number of features), and p (rank of the updated Hessian approximation) over 30 trials each in seconds.

Sensing Problem (Binary Classification)

Logistic Regression Problem

$$f_{\text{logreg}}(x) := -\frac{1}{p}\sum_{i=1}^{p} \log(\sigma(b_i a_i^T x))$$

- Labels $b_i \in \{1, -1\}$ with equal probability (class balanced)
- Decay rate $c_j = \exp(-\beta_j)$, $j = 1, ..., n$
- Noise decaying with feature $N_{ij} = z_{ij} c_j$, where $z_{ij} \sim \mathcal{N}(0, 1)$
- High signal regime $A_{ij} = b_i z_{ij} + N_{ij}$
- Low signal regime: $A_{ij} = b_i z_{ij}(1 - c_j) + N_{ij}$

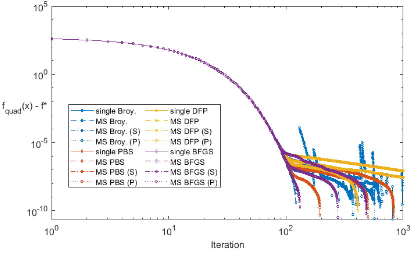


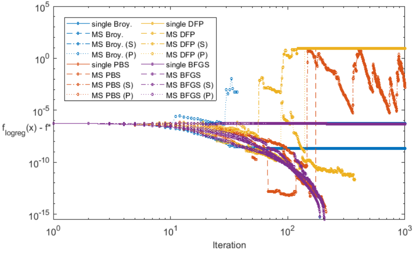
Quadratic vs non-Quadratic (Logsitic Regression)

$$f_{\mathrm{quad}}(x) \quad := \quad \frac{1}{2p}||Ax - b||_2^2,$$

$$f_{\mathrm{logreg}}(x) \quad := \quad -\frac{1}{p}\sum_{i=1}^{p}\log(\sigma(b_i a_i^T x)).$$

- Quadratic problems : Hessian and $Y^T S$ are always PSD
- Logistic regression : Not guaranteed to be descent step and unstable (sometimes diverge).





Simulation Results

- Average number of iterations to reach $f(x_k) - f^* \leq \epsilon := 10^{-9}$, over 10 trials.

	Low signal regime			High signal regime		
	Easy	Med.	Hard	Easy	Med.	Hard
Single Broyden	77.3	81.1	92.1	69.6	69.6	69.6
MS Broyden (Vanilla)	67.5	**65.9**	83.4	**55.7**	59.4	**51.6**
MS Broyden (Symmetric)	—	—	—	—	—	—
MS Broyden (PSD)	81.9	95.1	84.8	89.9	81.2	74.7
Our Broyden (Backsolve, B^{-1})	**60.1**	**65.9**	**71.8**	57.8	**54.6**	62.3
Single PSB	77.3	81.1	92.1	69.6	69.6	69.5
MS PSB (V)	80.5	78.0	83.4	62.3	64.1	57.9
MS PSB (S)	227.9	207.3	285.6	115.4	123.7	214.5
MS PSB (P)	**66.0**	**71.4**	**83.1**	**58.8**	**62.8**	**53.4**
Our PSB (B)	71.8	97.5	102.0	85.3	74.6	61.2
Single DFP	**77.3**	**81.1**	**92.1**	**69.6**	**69.6**	**69.6**
MS DFP (V)	133.5	169.6	151.6	108.1	128.6	107.7
MS DFP (S)	128.3	248.2	213.3	158.5	252.5	117.2
MS DFP (P)	94.3	103.8	114.7	105.7	81.7	110.8
Our DFP (B)	165.9	172.3	142.8	120.6	153.6	120.8
Single BFGS	77.3	81.1	92.1	69.6	69.6	69.6
MS BFGS (V)	69.9	84.9	72.7	67.6	57.3	59.5
MS BFGS (S)	249.7	182.9	296.6	144.2	192.0	252.6
MS BFGS (P)	76.5	75.0	92.4	64.0	68.7	76.3
Ours (B)	65.6	65.7	88.3	65.0	62.6	64.2
Ours (Woodbury)	**45.0**	**51.0**	**60.2**	**26.4**	**27.6**	**27.7**

- Failure rate (diverge or didn't converge in 500 iter) over 18 problems, 10 trials each

	single secant	(V)	(S)	(P)	our(B)	our(W)
Broyden	**0**	0.18	1.00	0.072	0.11	— (not executed)
PSB	**0**	0.028	0.67	**0**	**0.0056**	— (not executed)
DFP	**0**	0.039	0.79	0.028	0.050	— (not executed)
BFGS	**0**	**0.017**	0.68	**0**	**0.0056**	**0.017**

Conclusion and Future Direction

- Multisecant methods are potentially powerful but not popular because non-quadratic problems are not stable.
- Almost multisecant quasi-Newton BFGS method approximates curvature (second-order) information, sacrificing secant conditions but it guarantees descent direction (stable).
- Getting μ is computationally far cheaper to achieve PSD hessian compared to the singular value decomposition on the full matrix.
- Expand our methods to limited memory version : "Almost multisecant L-BFGS quasi-Newton method".
- Apply to non-convex problems such as Neural Network.

  Stochastic QN method, Sketching QN method, etc

Thank you