Almost multisecant BFGS quasi-Newton method

Mokhwa Lee
Applied Mathematics and Statistics
Stony Brook University
NY, USA
mokhwa.lee@stonybrook.edu

Yifan Sun
Computer Science
Stony Brook University
NY, USA
yifan.0.sun@gmail.com

***Abstract*—In convex optimization, quasi-Newton (QN) methods provide an alternative to second-order techniques for solving minimization problems by approximating curvature of a given target function. This approach reduces computational complexity as it relies solely on first-order information, and satisfies the secant condition. This paper focuses on multi-secant (MS) extensions of QN, which enhances the Hessian approximation at low cost for not only quadratic but also non-quadratic problems. Specifically, we use a low-rank perturbation strategy to construct an almost-secant QN method that maintains positive definiteness of the Hessian estimate, which in turn helps ensure constant descent direction and reduces method divergence for non-quadratic problems. Our results show that our perturbations can improve stability and effectiveness of multisecant updates.**

***Index Terms*—Quasi-Newton, Hessian approximation, low-rank approximation, positive semidefinite update, multisecant method, second order optimization, convex optimization**

I. Introduction

We consider the unconstrained minimization problem

$$\underset{x}{\text{minimize}} \quad f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex, twice-differentiable everywhere, and bounded below. Newton's method iteratively solves the linear system of order n to get a search direction d_k,

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k) \tag{2}$$

where $\nabla^2 f(x_k)$ is the Hessian and $\nabla f(x_k)$ is the gradient of the kth iterate. In this case, the next iterate is updated as

$$x_{k+1} = x_k + \alpha d_k$$

where $\alpha > 0$ is a step length parameter. However, while this method is foundational in continuous optimization, when dealing with large-scale problems, getting the Hessian matrix and solving (2) is not computationally scalable. For this reason, quasi-Newton (QN) methods, like BFGS (12), are introduced and become good substitutes which efficiently approximate the Hessian with simple operations performed on successive gradient vectors.

a) QN methods: The usual QN method involves forcing the satisfaction of a single-secant equation

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k). \tag{3}$$

where where $B_k \in \mathbb{R}^{n \times n}$ serves as a Hessian approximation of f at x_k, and the equation (3) is the 2nd-order Taylor approximation of f at x_k. The iterates are then updated via an approximate Newton step

$$x_{k+1} = x_k - \alpha B_k^{-1} \nabla f(x_k). \tag{4}$$

b) Multi-secant QN methods: A stronger, lesser-explored family of approximations are the *multisecant QN methods*, which satisfy

$$B_k(x_i - x_j) = \nabla f(x_i) - \nabla f(x_j) \tag{5}$$

for some subset of $i \neq j \in \{k, k-1, ..., k-p+1\}$ where p is the number of previous information taken into account. Conventionally, p is a small positive integer such that $p \ll n$; in our approach, we set p within the range of 5 to 15. While multisecant extensions have been explored in past literature (5), and are shown to be more powerful approximations than single-secant approaches, there are several key challenges that prevent them from mainstream usage:

1) The rank of the update is p (the number of secant conditions), and thus multisecant approaches have a constant factor overhead in computational complexity.
2) Multisecant versions of QN methods often struggle with stability. Specifically, in the case of DFP and BFGS, a single-secant update is guaranteed to be a "descent search direction"; however, incorporating multisecant conditions destroys this valuable descent property, and can lead to divergence.

While the first deficiency presents an annoyance, in general linear scaling of complexity is usually not prohibitive. However, the second deficiency is severe; for this reason, multisecant QN methods seem popular only in quadratic optimization, and are not easily generalizable even for convex functions. Therefore, in this paper we target the second deficiency, by presenting a method with nearly identical runtime as the usual multisecant QN updates, but with a perturbation that ensures descent directions. By exploiting important linear algebra properties, our strategy is low in computational cost, and adds the necessary perturbation to accelerate multisecant QN methods on important ill-conditioned problems.

A. Related works.

Perhaps the most well-known family of single-secant quasi-Newton methods are Broyden's method (1), Powell symmetric Broyden's (PSB) (11), Davidson-Fletcher-Powell (DFP) (7),

and BFGS named after the concurrent works of (1), (2), (3), and (4). These methods, though distinct, form a progression; Broyden's method is first, then Powell's method introduces symmetric updates, and DFP and BFGS simultaneously introduce positive semidefiniteness (PSD) in B_k. These qualities (symmetric and PSD) are often desired to ensure that $d_k = -B_k^{-1}\nabla f(x_k)$ is indeed a descent direction; these advanced methods are often more stable in practice.

The *multisecant* extensions were first explored not long later; (5) for Broyden's method, and (6) for extensions of Broyden's, Powell's method, DFP, and BFGS updates. These methods also attempt to progressively include desired features, such as 1. fast and cheap updates, 2. symmetry, and 3. positive definiteness. However, the addition of these features is much less straightforward in the multisecant case; for this reason, multisecant methods are primarily used to solve quadratic systems, where symmetric PSD updates of multisecant DFP and BFGS are easier to guarantee. *However, for general convex optimization problems, multisecant quasi-Newton methods do not ensure descent.*

An important extension, L-BFGS (9), is a limited memory version of the BFGS algorithm and is widely used in machine learning. Additionally, Fang and Saad (8) also proposed the generalization of Broyden's and Multisecant family with several successful techniques for handling QN-type problems. More recently, closely related works include (13), (14), and (15). These are higher rank update schemes that use only first-order information, and are shown to achieve q-superlinear convergence (18), at least in the local sense.

In this work, we explore various techniques to impose symmetric and PSD updates in multisecant quasi-Newton methods through carefully tuned perturbations, for *ill-conditioned non-quadratic problems*. We compare these techniques against the perturbation methods presented in the seminal work (6).

II. Preliminaries

Consider the goal of identifying B_k such that the following linear equations are satisfied

$$B_k(x_i - x_j) = \nabla f(x_i) - \nabla f(x_j), \quad i, j \leq k. \tag{6}$$

If $x_k \in \mathbb{R}^n$, then a symmetric Hessian estimate $B_k \in \mathbb{R}^{n\times n}$ has more than one degrees of freedom if $n > 1$. In other words, for a reasonable number of secant constraints, the system of equation (6) is under-determined, thus leading to the construction of a variety of QN methods. We first present a brief overview of four historically important methods, first in the single-secant framework, and then the multisecant generalization.

A. Single-secant quasi-Newton equations

In this section, the Hessian approximation B_{k+1} satisfies the single-secant condition

$$B_{k+1}\underbrace{(x_{k+1} - x_k)}_{s_k} = \underbrace{\nabla f(x_{k+1}) - \nabla f(x_k)}_{y_k} \tag{7}$$

which is derived from the Taylor's second order expansion and its differential

$$\nabla f(x_{k+1}) \approx \nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k)$$

where $B_{k+1} \approx \nabla^2 f(x_{k+1})$. If we restrict B_{k+1} to be symmetric, then the equation (7) has $\frac{n(n+1)}{2}$ variables but only n constraints to define B_{k+1}, where $n \geq 1$. If $n = 1$, then (7) has a unique solution; however, when $n > 1$, there are many variations of quasi-Newton methods that equally satisfy (7) because the system is under-determined. After computing B_{k+1}, each quasi-Newton method will update the next iterate x_{k+2} as

$$x_{k+2} = x_{k+1} - \alpha B_{k+1}^{-1}\nabla f(x_{k+1}).$$

To guarantee that each step taken is in a descent direction, the following

$$-\nabla f_k^T B_k^{-1}\nabla f_k < 0 \tag{8}$$

should be satisfied for all k. If B_k is not PSD, (8) is not necessarily satisfied and hence the algorithm will not be guaranteed to monotonically decrease at each iteration. Therefore, maintaining B_k PSD is an important key for QN methods.

B. Single-secant quasi-Newton methods

The most well-known four single-secant QN methods are described below. First, the Broyden's update (10) is given as

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

Here, we generally start with $B_0 = I$. It is a simple exercise to show that $B_{k+1}s_k = y_k$ by multiplying s_k on the left and the right hand side. However, there is no guarantee that such an update will maintain B_k to be symmetric or PSD. This is done by a series of follow-up updates; Powell's (11) gives symmetry

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

and DFP (7) and BFGS (1; 2; 3; 4) give symmetry and PSD:

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

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

It has previously been shown that DFP and BFGS achieve q-superlinear convergence (16).

C. Multi-secant quasi-Newton methods

Equation (6) explained four typical multisecant QN methods. Firstly, we consider two choices for s_i and y_i: the "curve-hugging" version for $i = k, ..., k-p+1$ such that

$$s_i = x_{i+1} - x_i, \quad y_i = \nabla f(x_{i+1}) - \nabla f(x_i) \tag{9}$$

and the "anchored at most recent" version for $i = k-1, ..., k-p$ such that

$$s_i = x_{k+1} - x_i, \quad y_i = \nabla f(x_{k+1}) - \nabla f(x_i). \tag{10}$$

The two interpolating schemes have different benefits. For the simplicity, we will use the former "curve-hugging" version. We represent these choices with matrices S_k and Y_k as

$$S_k = \begin{bmatrix} | & | & & | \\ s_{k-p} & s_{k-p+1} & \dots & s_k \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times p} \tag{11}$$

$$Y_k = \begin{bmatrix} | & | & & | \\ y_{k-p} & y_{k-p+1} & \dots & y_k \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times p} \tag{12}$$

where $s_i = x_{i+1} - x_i$ and $y_i = \nabla f(x_{i+1}) - \nabla f(x_i)$. Then, we can define multisecant (MS) condition

$$B_{k+1}S_k = Y_k \tag{13}$$

which interpolates p number of previous iterates. Given the matrices S_k and Y_k, the number of free variables to identify B_{k+1} in (13) is still $n(n-1)/2 > pn$, so (13) is still an underdetermined problem when $p < n/2$ (usual regime). Reference (6) presented the following four multisecant generalizations of QN methods:

$$B_{k+1} = B_k + (Y_k - B_kS_k)(S_k^TS_k)^{-1}S_k^T \quad \text{(MS Broyden)}$$

$$\begin{aligned} B_{k+1} = {} & B_k + (Y_k - B_kS_k)(S_k^TS_k)^{-1}S_k^T \\ & + S_k(S_k^TS_k)^{-1}(Y_k - B_kS_k)^T \\ & - S_k(S_k^TS_k)^{-1}(Y_k - B_kS_k)^TS_k(S_k^TS_k)^{-1}S_k^T \end{aligned} \quad \text{(MS PSB)}$$

$$\begin{aligned} B_{k+1} = {} & B_k + (Y_k - B_kS_k)(Y_k^TS_k)^{-1}Y_k^T \\ & + Y_k(Y_k^TS_k)^{-1}(Y_k - B_kS_k)^T \\ & - Y_k(Y_k^TS_k)^{-1}(Y_k - B_kS_k)^TS_k(Y_k^TS_k)^{-1}Y_k^T \end{aligned} \quad \text{(MS DFP)}$$

$$\begin{aligned} B_{k+1} = {} & B_k + Y_k(Y_k^TS_k)^{-1}Y_k^T \\ & - B_kS_k(S_k^TB_kS_k)^{-1}S_k^TB_k \end{aligned} \quad \text{(MS BFGS)}$$

In the reference (6), it is noted that Powell's B_{k+1} is guaranteed to be symmetric only if $S_k^TY_k$ is symmetric, and DFP's and BFGS's B_{k+1} is symmetric+PSD only if $Y_k^TS_k$ is symmetric+PSD. To see this, note that the MS constraint (13) enforces $S_k^TB_{k+1}S_k = S_k^TY_k$, so the symmetry / PSD-ness of B_{k+1} is not possible if $S_k^TY_k$ does not have the same corresponding properties. *However, this assumption is too strong for general convex functions f. In fact, outside of f being a quadratic function, it is usually untrue.* In (6), the problem is addressed using perturbations of an estimated Cholesky factorization of B_k. Here, we investigate simple diagonal perturbations.

TABLE I: Quasi-Newton method comparison. Our method, applied on any of the multisecant QN methods, sacrifices the multisecant condition for PSD.

Method	Symm.	PSD	MS cond.	update rank
Broyden's	×	×	×	1
PSB	✓	×	×	3
DFP	✓	✓	×	3
BFGS	✓	✓	×	2
Multisecant QN	×	×	✓	$2p$
Ours	✓	✓	≈	$2p$

D. Update of B_k^{-1}

A key feature of a successful quasi-Newton method is avoiding inverting matrices, or solving full linear systems, at each iteration. Therefore, it is desirable to have a closed-form update of the matrix inverse at each step. In the unique case of MS BFGS, we can update B_{k+1}^{-1} effectively using the "Sherman–Morrison–Woodbury formula (17)",

$$(B + UCV)^{-1} = B^{-1} - B^{-1}U(C^{-1} + VB^{-1}U)^{-1}VB^{-1} \tag{14}$$

where

$$B = B_k, \quad U = \begin{bmatrix} Y_k & B_kS_k \end{bmatrix}, \quad V = \begin{bmatrix} Y_k^T \\ S_k^TB_k \end{bmatrix},$$

$$C = \begin{bmatrix} (Y_k^TS_k)^{-1} & 0 \\ 0 & -(S_k^TB_kS_k)^{-1} \end{bmatrix}.$$

Therefore, we may write succinctly for $H_k = B_k^{-1}$, and iteratively update the approximate Hessian (MS BFGS inverse) H_{k+1} by Woodbury formula as follows:

$$\begin{aligned} H_{k+1} = {} & H_k - \\ & \begin{bmatrix} H_kY_k & S_k \end{bmatrix} \begin{bmatrix} Y_k^TS_k + Y_k^TH_kY_k & Y_k^TS_k \\ S_k^TY_k & 0 \end{bmatrix}^{-1} \begin{bmatrix} Y_k^TH_k \\ S_k^T \end{bmatrix}. \end{aligned}$$

Importantly, the term B_k is never needed in the update for MS BFGS inverse, however, this is not true for Broyden's, PSB, or DFP methods; that is, B_k is not canceled out in the process of getting H_{k+1} from B_{k+1} in their Woodbury inversions. Therefore, in these three methods, backsolve is still required at each iteration.

III. An almost-multisecant method

We first summarize all the existing multisecant QN methods in the form of

$$B_{k+1} = B_k - D_1W^{-1}D_2^T \tag{15}$$

with some $D_1 \in \mathbb{R}^{n\times p}$, $D_2 \in \mathbb{R}^{p\times n}$, $W \in \mathbb{R}^{p\times p}$, specified in Table II with different p values. (Note that W is not assumed to be symmetric nor PSD in general.) The natural perturbation to enforce symmetry and positive semidefiniteness is to add the additional term μI,

$$B_{k+1} = B_k - \frac{D_1W^{-1}D_2^T + (D_1W^{-1}D_2^T)^T}{2} + \mu I \tag{16}$$

TABLE II: All four multisecant (MS) methods can be written in form (15), with the following choices of D_1, D_2, and W, where W is related to the Schur complement of the update matrix. Here, we write $Z_k = Y_k - B_k S_k$. We also show the update of $H_k = B_k^{-1}$ for BFGS. *= update is consistent only if B_k is symmetric PSD at each iteration.

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

where μ is the smallest positive value needed to ensure that B_{k+1} to be positive semidefinite ($B_{k+1} \succeq 0$). That is, defining

$$\Delta = \mu I - \frac{1}{2}\begin{bmatrix} D_1 & D_2 \end{bmatrix}\begin{bmatrix} 0 & W_k^{-1} \\ W_k^{-T} & 0 \end{bmatrix}\begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix} \in \mathbb{R}^{n\times n}$$

then the goal is to find $\mu = \max\{0, -\lambda_{\min}(B_k + \Delta)\}$. Note that the multisecant condition $B_{k+1}S_k = Y_k$ cannot be exact when we perturb B_{k+1}, and this is the reason of being an "almost multisecant" scheme.

However, in general, finding $\lambda_{\min}(B_k)$ may not be computationally cheap. The obvious approach is to use a fast power method or Lanczos method, but there is no reason to assume that B_k is sparse, nor low rank after n iterations. Therefore, we assume that this operation is prohibitive, or at least can only be used rarely. Instead, we use low-rank operations to simply find $\mu = \max\{0, -\lambda_{\min}(\Delta)\}$. This is an overapproximation, and guarantees that B_{k+1} is symmetric and PSD. In cases where μ is too large, at worst the method behaves like gradient descent. However, in our numerical results, our method always outperforms gradient descent.

We now provide the main theorem that describes how to compute μ using only low rank updates by Schur Complement. Note that A, c, and F are explicitly chosen such that $H_1 = \Delta$ in the following Theorem 1.

Theorem 1. *Consider W a nonsymmetric matrix, and*

$$\Delta = \mu I - \frac{1}{2}\begin{bmatrix} D_1 & D_2 \end{bmatrix}\begin{bmatrix} 0 & W^{-1} \\ W^{-T} & 0 \end{bmatrix}\begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix}. \tag{17}$$

We take the eigenvalue decomposition of the $p \times p$ matrix $W^{-1} = U\Sigma V^T$, and for any $c > 0$, $0 \le \epsilon \le 1$, construct

$$\begin{aligned} &\Sigma = (S^2 - c^2 I)^{-1} S, && F = \frac{c\epsilon}{c + \|W\|_2} V S U^T, \\ &P = (cI - c^{-1} F F^T)^{-1}, && Q = (cI - c^{-1} F^T F)^{-1}, \\ &A = \begin{bmatrix} D_1 & D_2 \end{bmatrix}\begin{bmatrix} P & -cQF^T - W^{-1} \\ -cFQ - W^{-T} & Q \end{bmatrix}\begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix}. \end{aligned} \tag{18}$$

Then Δ is PSD if and only if

$$H_2 = \begin{bmatrix} cI & F \\ F^T & cI \end{bmatrix} - \begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix}(A + 2\mu I)^{-1}\begin{bmatrix} D_1 & D_2 \end{bmatrix}$$

is PSD.

Proof of Thm. 1. First, we verify that by this construction,

$$A = \begin{bmatrix} D_1 & D_2 \end{bmatrix}\begin{bmatrix} P & -cQF^T - W^{-1} \\ -cFQ - W^{-T} & Q \end{bmatrix}\begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix}.$$

is PSD. Based on our construction, A can be written as

$$\begin{aligned} A &= \begin{bmatrix} D_1 & D_2 \end{bmatrix}\begin{bmatrix} P & -cFQ - W^{-1} \\ -cQF^T - W^{-T} & Q \end{bmatrix}\begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix} \\ &= \begin{bmatrix} D_1 U & D_2 V \end{bmatrix}\underbrace{\begin{bmatrix} R & -c\delta SR - \Sigma \\ -c\delta RS - \Sigma & R \end{bmatrix}}_{:= E \in \mathbb{R}^{2p\times 2p}}\begin{bmatrix} U^T D_1^T \\ V^T D_2^T \end{bmatrix} \end{aligned}$$

where

$$R = (cI - c^{-1}\delta^2 S^2)^{-1}, \qquad \delta = \frac{c\epsilon}{c + \|W\|_2}.$$

Also, from (18), the first equation is a quadratic equation involving S_{ii} satisfying

$$S_{ii} = \frac{1 + \sqrt{1 + 4\Sigma_i^2 c^2}}{2\Sigma_i} \le \frac{1}{\Sigma_i} + c.$$

Note that since Σ_i are the singular values of W^{-1}, $\frac{1}{\Sigma_i} \le \|W\|_2$, which implies $S_{ii} \le \|W\|_2 + c$.

We are left to show if E is PSD. Note that we may partition E into 4 blocks of diagonal matrices, which means there exists a permutation PEP^T which is block diagonal, with 2-by-2 symmetric sub-blocks

$$E_{ii} = \begin{bmatrix} (c - \frac{1}{c}\delta^2 S_{ii}^2)^{-1} & \frac{\delta S_{ii}}{\delta^2 S_{ii}^2 - c^2} - \Sigma_{ii} \\ \frac{\delta S_{ii}}{\delta^2 S_{ii}^2 - c^2} - \Sigma_{ii} & (c - \frac{1}{c}\delta^2 S_{ii}^2)^{-1} \end{bmatrix} \in \mathbb{R}^{2\times 2}.$$

The (1,1) and (2,2) blocks can be shown to be positive since

$$\delta S_{ii} \le \frac{c}{c + \|W\|_2}(\|W\|_2 + c) = c. \tag{19}$$

Therefore, E_{ii} is PSD if and only if the (2,1) element has magnitude smaller than both diagonal elements; that is,

$$E_{ii} \succeq 0 \iff \frac{1}{c - \frac{1}{c}\delta^2 S_{ii}^2} \geq \frac{\delta S_{ii}}{\delta^2 S_{ii}^2 - c^2} - \Sigma_{ii}.$$

Since (19), this is equivalent to

$$c \geq -c_3 S_{ii} - \underbrace{(c^2 - c_3^2 S_{ii}^2)}_{\geq 0} \Sigma_{ii}$$

which is true since the right hand side is negative.

Next, we consider the matrix H

$$H = \begin{bmatrix} A + 2\mu I & D_1 & D_2 \\ D_1^T & cI & F \\ D_2^T & F^T & cI \end{bmatrix}$$

where c is a non-negative scalar, F is a $p \times p$ matrix (yet undefined), and A is some (unspecified) symmetric matrix. Then the two Schur complements of H are H_1 and H_2:

$$H_1 := A + 2\mu I - \begin{bmatrix} D_1 & D_2 \end{bmatrix} \begin{bmatrix} cI & F \\ F^T & cI \end{bmatrix}^{-1} \begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$$H_2 := \begin{bmatrix} cI & F \\ F^T & cI \end{bmatrix} - \begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix} (A + 2\mu I)^{-1} \begin{bmatrix} D_1 & D_2 \end{bmatrix} \in \mathbb{R}^{2p \times 2p}$$

Then,

$$H_1 \text{ is PSD and } \begin{bmatrix} cI & F \\ F^T & cI \end{bmatrix} \text{ is PSD and invertible}$$

if and only if

$$H_2 \text{ is PSD and } A + 2\mu I \text{ is PSD and invertible.}$$

Since we already showed that A is PSD, we see that testing H_2 for PSD is sufficient to guarantee H_1 is PSD.

□

This proposes a method for choosing μ, which is summarized in Alg. 1 with the computation time in TABLE III; the full method is summarized in Alg. 2.

TABLE III: Performance metrics of Alg. 1 with varying m (number of data), n (features), and p (past information), over 30 trials each in seconds.

m	n	p	AvgElapsedTime	StdElapsedTime
10000000	10000	5	1.342049018	0.198843912
10000000	10000	10	3.636063824	0.780097481
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

Algorithm 1 Compute μ

Input: D_1, D_2, W, $i_{\max}$
Output: μ such that Δ in (17) is PSD.

$[U, \Sigma, V] = \mathbf{svd}(W)$
Define F, P, Q and S as in (18).
$\mu = 0.01$
Construct $\bar{D} = \begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix} \begin{bmatrix} D_1 & D_2 \end{bmatrix}$ and

$$C = \begin{bmatrix} P & -cQF^T - W^{-1} \\ -cFQ - W^{-T} & Q \end{bmatrix}.$$

for $i = 1$ **to** $i_{\max}$ **do**
  Find H_2 where $A + 2\mu I$ is inverted by Woodbury (14),

$$H_2 = \begin{bmatrix} cI & F \\ F^T & cI \end{bmatrix} - \frac{1}{2\mu}\bar{D} + \frac{1}{2\mu}\bar{D}(2\mu C^{-1} + \bar{D})^{-1}\bar{D}$$

  if $\lambda_{\min}(H_2) > 10^{-15}$ **then**
    break.
  else
    $\mu \leftarrow 2\mu$
  end if
end for

Algorithm 2 Almost multisecant quasi-Newton algorithm

Input: $B, x_0, \alpha, p, f(x), \nabla f(x)$
Output: $f_{k+1}(x)$

$B_0 = I$
for $k = 1, \ldots, T$ **do**
  Update s_1,...,s_p, and $y_1, ..., y_p$ using (9) or (10).
  Update S_k, Y_k using (11) and (12).
  Compute D_1, D_2, W
  Use Alg. 1 to pick μ
  Update B_{k+1}^{-1} using (16)
  Update $x_{k+1} = x_k - \alpha B_{k+1}^{-1} \nabla f(x_k)$
end for

IV. Numerical Results

A. Compute μ

Table IV shows the runtime of Algorithm 1 vs eigenvalue decompositions using full (`eig`) or fast partial (`eigs`) operations. By leveraging low-rank structure, we significantly reduce the runtime complexity.

B. Sensing problem

Our experiments are performed over a sensing problem that is tuned to make the Hessian ill-conditioned; that is, problems in which quasi-Newton methods (and especially MS variants) perform better than gradient or single-secant methods.

Our sensing model is $\mathbf{A}x \approx \mathbf{b}$, where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, and $b \in \mathbb{R}^m$. We construct each variable (parameter) as follows:

- Labels $\mathbf{b}_i \in \{1, -1\}$ with equal probability (class balanced)

TABLE IV: Runtime (in seconds) comparison of Alg. 1, vs direct eigenvalue computation. Format: mean(std).

n	eig	eigs	Alg. 1 $i_{\max}=10$	Alg. 1 $i_{\max}=30$	Alg. 1 $i_{\max}=50$
500	3.3e-2 (1.1e-2)	2.0e-3 (8.0e-4)	7.2e-4 (2.6e-4)	7.3e-4 (2.7e-4)	6.6e-4 (2.5e-4)
1000	8.9e-2 (3.0e-2)	4.8e-3 (2.0e-3)	1.7e-3 (5.8e-4)	1.8e-3 (6.4e-4)	1.9e-3 (6.3e-4)
2500	5.8e-1 (1.9e-1)	3.1e-2 (1.0e-2)	9.6e-3 (3.2e-3)	9.8e-3 (3.3e-3)	1.0e-2 (3.5e-3)
5000	2.3 (7.6e-1)	2.1e-01 (7.0e-2)	3.6e-2 (1.2e-02)	3.6e-2 (1.2e-2)	3.8e-02 (1.3e-2)

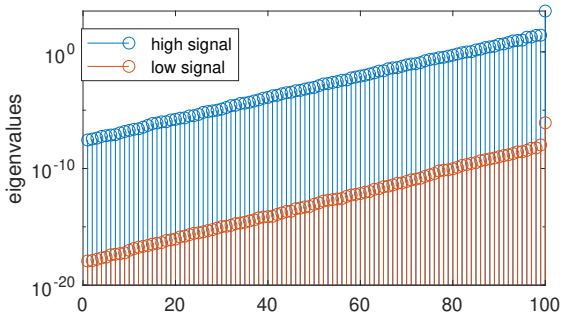

Fig. 1: Spectrum of Hessian in sensing problem

- Decay rate $c_j = \exp(-\beta_j), j = 1, ..., n$
- Noise decaying with feature

$$\mathbf{N}_{i,j} = z_{i,j} c_j, \qquad z_{i,j} \sim \mathcal{N}(0,1)$$

- High signal regime $\mathbf{A}_{i,j} = \mathbf{b}_i z_{i,j} + \mathbf{N}_{i,j}$
- Low signal regime: $\mathbf{A}_{i,j} = \mathbf{b}_i z_{i,j}(1 - c_j) + \mathbf{N}_{i,j}$

The spectrum of the Hessian in logistic regression, using these problem parameters and objective function (21) is shown in Fig 1.

C. Quadratic vs non-Quadratic

We compare the performance of the proposed MS-BFGS variants in Figure 2, where the quadratic and logistic regression problems minimize the following respective cost functions:

$$f_{\text{quad}}(x) := \frac{1}{2m}||\mathbf{A}x - \mathbf{b}||_2^2, \tag{20}$$

$$f_{\text{logreg}}(x) := -\frac{1}{m}\sum_{i=1}^{m} \log(\sigma(\mathbf{b}_i \mathbf{a}_i^T x)). \tag{21}$$

where σ is a sigmoid function defined as $\sigma(z) = \frac{1}{1+\exp(-z)}$. Specifically, for quadratic problems, while the variations and enhancement do offer improvements, up to a pretty small error most enhancements are relatively irrelevant, as $Y^T S$ is always positive semidefinite (PSD) and symmetrization / perturbations are not needed to make MS methods stable. On the other hand, for logistic regression problems (non-quadratic), non-perturbed MS methods are not guaranteed to be stable; they do often diverge.

D. Logistic regression

We now give more extensive results for minimizing the logistic regression problem (21). Here, we construct the problem specifically to make optimizing f difficult without second-order information. We use $\alpha = 0.1$ as the backoff parameter in all cases.

We compare the performance over the four base QN methods: Broyden's, PSB, DFP, and BFGS. In each case, we compare over five variants: single-secant ($p = 1$) update, vanilla MS update (V), symmetrization-only update (where $\mu = 0$ in (16)) (S), PSD-direct update (where μ is computed exactly as the smallest perturbation to make B_{k+1}^{-1} PSD), and ours (Algorithm 2). In the case of BFGS, we also include an update where the entire procedure is replaced with perturbing $H_k = B_k^{-1}$, rather than B_k. We differentiate the two as (B) (for backsolve) and (W) (for Woodbury). Note that we obtained μ by the Theorem 1 and the equation (16). However, in practice, to enhance stability and performance, we simply add μI to equation (15) in the formulation of our method (Ours (W)).

Table VI gives the average number of iterations to reach the tolerance under an easy ($\beta = 10/n$), medium (($\beta =$

TABLE V: Failure rate (diverge or did not converge in 500 iterations) over 18 problems, 10 trials each.

	sin.	(V)	(S)	(P)	our(B)	our(W)
Broyden	0	0.18	1.00	0.072	0.11	–
PSB	0	0.028	0.67	0	0.0056	–
DFP	0	0.039	0.79	0.028	0.050	–
BFGS	0	0.017	0.68	0	0.0056	0.017

TABLE VI: 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 Broy.	77.3	81.1	92.1	69.6	69.6	69.6
MS Broy. (V)	67.5	**65.9**	83.4	**55.7**	59.4	**51.6**
MS Broy. (S)	–	–	–	–	–	–
MS Broy. (P)	81.9	95.1	84.8	89.9	81.2	74.7
Our Broy. (B)	**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 (W)	**45.0**	**51.0**	**60.2**	**26.4**	**27.6**	**27.7**

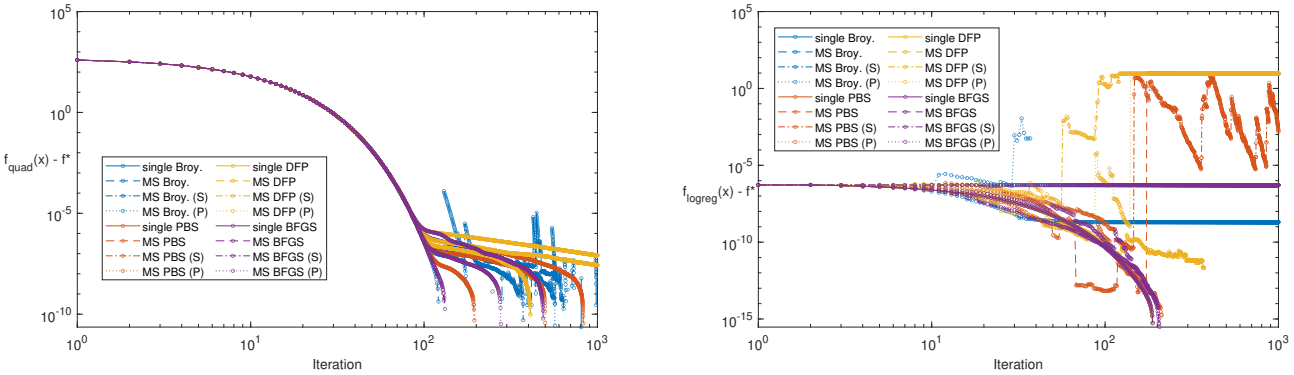

Fig. 2: Comparison of QN methods (single and multisecant) on quadratic (20) vs logistic regression (low signal) (21) with $p = 5$.

$20/n$), and hard ($\beta = 30/n$) regime. In several cases, the method diverged or did not converge in 500 iterations, and was labeled a "failure"; the failure rate across all experiments is given in Table V. Interestingly, direct symmetrization, without ensuring PSD, almost certainly guarantees failure, highlighting the delicacy of these methods; careless perturbations will not fail gracefully. We observe the following trends.

1) In Broyden's method and BFGS, our perturbation often performed best, and in the BFGS case, with the Woodbury inversion, significantly outperformed all other MS-QN methods.
2) In PSB methods, the best method is an exact computation of μ, followed by our approach. This shows that we have the right idea, but Algorithm 1 is a bit noise-prone. This is not surprising, since doubling μ at each iteration means that the final estimate is often very coarse. Note also that MS BFGS(P) in Table VI is not a feasible method in practice, as it requires full spectral computations; it is only a mark of comparison.
3) In DFP, it does not look like any method is more superior than the single-secant variant. In fact, all multi-secant approaches seem to deteriorate performance significantly. This is perhaps because the family of logistic regression problems we have chosen are not the best fit for DFP.

To clarify, the method (P) adds a $-\lambda_{\min}(H)I$ term to the approximate Hessian, which is like an "oracle" because in general $\lambda_{\min}(H)$ cannot be computed at each iteration. These rows are included as unrealistic baselines. We also note that in terms of actual implementation practically, we observe that only BFGS can be useful because the others cannot update their inverse Hessian approximates in closed form, and require linear system solves at each iteration. Overall, though, Table VI is meant to be comprehensive, and the four QN variants do not always behave identically.

V. Conclusion

We explored multisecant quasi-Newton methods as fast curvature-approximating methods. The time complexity of all methods is $O(n^2)$ for the matrix-vector multiplications; this is the same for all multiscant QN methods. However, Our key contribution is in the *number of steps to convergence* which is controlled by ensuring descent at each step. Specifically, we look at convex nonquadratic problems, where, because of the multisecant feature, the step directions are not always descending (Hessian approximate is not always positive semidefinite), which leads to method instability. We propose a unified fast perturbation strategy in terms of a small μ perturbation, whose update is computed efficiently, and approximates the update steps for Broyden's, Powell's, DFP, and BFGS, which introduce little overhead but offer improved convergence and stability.

References

[1] C. G. Broyden. "The convergence of a class of double-rank minimization algorithms 1. General considerations." IMA Journal of Applied Mathematics, Volume 6, Issue 1, March 1970, Pages 76–90.

[2] R. Fletcher. "A new approach to variable metric algorithms." The Computer Journal, Volume 13, Issue 3, 1970, Pages 317–322.

[3] Goldfarb, Donald. "A family of variable-metric methods derived by variational means." Mathematics of Computation 24, no. 109 (1970): 23–26.

[4] Shanno, D. F. "Conditioning of quasi-Newton methods for function minimization." Mathematics of Computation 24, no. 111 (1970): 647–56.

[5] David M. Gay, Robert B. Schnabel. "Solving systems of nonlinear equations by Broyden's method with projected updates." Nonlinear Programming 3, Academic Press, 1978, Pages 245-281.

[6] Schnabel, Robert B. "Quasi-Newton methods using multiple secant equations." Computer Science Technical Reports 244.41 (1983): 06.

[7] Davidon, William C. "Variable metric method for minimization." SIAM Journal on optimization 1.1 (1991): 1-17.

[8] Fang, Haw-ren, and Yousef Saad. "Two classes of multisecant methods for nonlinear acceleration." Numerical linear algebra with applications 16.3 (2009): 197-221.

[9] Liu, Dong C., and Jorge Nocedal. “On the limited memory BFGS method for large scale optimization.” Mathematical programming 45.1 (1989): 503-528.

[10] Broyden, Charles G. “A class of methods for solving nonlinear simultaneous equations.” Mathematics of computation 19.92 (1965): 577-593.

[11] Abd Alamer, Mohammed, and Saad Mahmood. “On positive definiteness of Powell symmetric Broyden (H-version) update for unconstrained optimization.” AIP Conference Proceedings. Vol. 2834. No. 1. AIP Publishing, 2023.

[12] Dennis, Jr, John E., and Jorge J. Moré. “Quasi-Newton methods, motivation and theory.” SIAM review 19.1 (1977): 46-89.

[13] Gao, Wenbo, and Donald Goldfarb. “Block BFGS methods.” SIAM Journal on Optimization 28.2 (2018): 1205-1231.

[14] Liu, Chengchang, Cheng Chen, and Luo Luo. “Symmetric rank-k methods.” arXiv preprint arXiv:2303.16188 (2023).

[15] Mokhtari, Aryan, Mark Eisen, and Alejandro Ribeiro. “IQN: An incremental quasi-Newton method with local superlinear convergence rate.” SIAM Journal on Optimization 28.2 (2018): 1670-1698.

[16] Dennis, John E., and Jorge J. Moré. “A characterization of superlinear convergence and its application to quasi-Newton methods.” Mathematics of computation 28.126 (1974): 549-560.

[17] Woodbury, Max A. “Inverting modified matrices.” Department of Statistics, Princeton University, 1950.

[18] Rodomanov, Anton, and Yurii Nesterov. “Greedy quasi-Newton methods with explicit superlinear convergence.” SIAM Journal on Optimization 31.1 (2021): 785-811.