Quasi-Newton Methods and BFGS

CS427 Mathematics for Data Science: Project

Om Patil (200010036)
Hrishikesh Pable (200010037)
Chidaksh Ravuru (200010046)

April 2023

Contents

1 Abstract

In this project we present different optimization algorithms used for minimizing a given objective function. The focus is on gradient descent, Newton's method, and quasi-Newton methods such as BFGS. The aim of this project is to provide a comprehensive overview of these optimization algorithms, highlighting their advantages and disadvantages.

We begin by introducing the basic concepts of optimization and discussing the key properties of objective functions that make them amenable to optimization. Then, the gradient descent algorithm is introduced. Next, the Newton's method is introduced, and its advantages and disadvantages are discussed. This leads to the introduction of quasi-Newton methods, which aim to overcome the computational cost of computing the Hessian matrix in Newton's method. The focus is on the BFGS algorithm, which is widely used due to its good convergence properties and low computational cost.

2 Gradient Descent

The Gradient Descent Method is a commonly used optimization technique in machine learning and statistics. It is a first-order optimization algorithm that can be used to minimize the given function. Gradient Descent or Steepest Descent is a line search method where the decent direction along which we move is opposite to the gradient at that point. The Algorithm of Gradient Descent is Mentioned below.

2.1 Algorithm

Algorithm 1 Gradient Descent

Require: Initial guess $x^{(0)}$, step size $\alpha > 0$, stopping criterion $\left\|\nabla f(x^{(k)})\right\|_2 < \epsilon$, for some $\epsilon > 0$

1: $k \leftarrow 0$
2: **while** $\left\|\nabla f(x^{(k)})\right\|_2 > \epsilon$ **do**
3: Compute gradient $\nabla f(x^{(k)})$
4: Update: $x^{(k+1)} \leftarrow x^{(k)} - \alpha \nabla f(x^{(k)})$
5: $k \leftarrow k + 1$
6: **end while**
7: **return** $x^{(k)}$

3 Newton Method for Optimization

Newton's method is an iterative algorithm that is commonly used to solve optimization problems. In particular, it is widely used in convex optimization,

where the goal is to minimize a convex function subject to some constraints.

3.1 Algorithm

The basic idea behind Newton's method is to use a second-order approximation of the objective function in order to iteratively update the current estimate of the minimum. The algorithm can be summarized as follows:

1. Initialize x_0.
2. For $k = 0, 1, 2, ...$:
   (a) Compute the gradient $\nabla f(x_k)$ and the Hessian matrix $H_f(x_k)$.
   (b) Solve the system of linear equations $H_f(x_k)d_k = -\nabla f(x_k)$ for d_k.
   (c) Update the estimate: $x_{k+1} = x_k + \alpha_k d_k$, where α_k is chosen via a line search.
3. Return x_k as the final estimate.

3.2 How did we arrive at the search direction

Our main objective is to min $f(x)$ s.t $x \in R^n$

Out of many search directions along which the function decreases (we have seen steepest descent direction (p_k) which is along $-\nabla f_k$) another important search direction along which the function decreases is newton's direction. From second-order approximation of Taylor's series,

$$m_k(p) = f(x_k + p) \approx f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p \tag{1}$$

we want to minimize the above equation w.r.t p so by differentiating with respect to p and equating it to zero, we get:

$$\nabla f_k + 2 \times \frac{1}{2} \nabla^2 f_k p_k = 0 \implies p_k = -(\nabla^2 f_k)^{-1} \nabla f_k$$

The above search direction p_k is known as **Newton's direction**.

3.3 Assumptions for Netwon's method to converge to a minima

1. One Assumption we made in the above proof is that the function f is $\in C^2$ and the **Hessian $\nabla^2 f_k$ is positive definite.**

$$\nabla f_k^T p_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k^T \nabla f_k \tag{2}$$

   The above equation should be less than zero for p_k to be search direction for minimizing the function (also called descent direction) and converging to minima. Hence we need the Hessian to be positive definite.

2. For the Taylor Series Approximation we made, **if $||p||$ is small, the approximation of $f(x_k + p)$ is quite accurate**.

   As we assumed Hessian is $\in C^2$ (in the above assumption), we can say that Hessian is sufficiently smooth as it has continuous first and second-order partial derivatives.

   We want to show that the perturbation introduced in Newton's method due to the replacement of $\nabla^2 f(x_k + tp)$ with $\nabla^2 f_k$ in the expansion of $f(x_k + p)$ is of order $o(p^3)$.

$$\begin{aligned} f(x_k + p) &= f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T \nabla^2 f(x_k) p + O(|p|^3) \\ &= m_k(p) + \frac{1}{2} p^T (\nabla^2 f(x_k + tp) - \nabla^2 f(x_k)) p + O(|p|^3) \end{aligned}$$

   Since $\nabla^2 f$ is sufficiently smooth, we can Taylor expand it around x_k:

$$\begin{aligned} \nabla^2 f(x_k + tp) &= \nabla^2 f(x_k) + t \nabla^3 f(x_k)^T p + O(t^2 |p|^2) \\ &= \nabla^2 f(x_k) + O(|p|) \end{aligned}$$

   Plugging this into the previous equation:

$$\begin{aligned} f(x_k + p) &= m_k(p) + \frac{1}{2} p^T (\nabla^2 f(x_k) + O(|p|) - \nabla^2 f(x_k)) p + O(|p|^3) \\ &= m_k(p) + O(|p|^3) \end{aligned}$$

   Therefore, the difference between $f(x_k + p)$ and $m_k(p)$ is bounded by $O(|p|^3)$, which shows that when p is small, the approximation $f(x_k + p) \approx m_k(p)$ is quite accurate.

3.4 Why Newton when we have Gradient Descent

Newton's method has several advantages over other optimization algorithms, such as gradient descent. For one, it converges more quickly (typically Quadratic) to the optimum, especially when the objective function is strongly convex. Additionally, it can handle constraints more easily than other algorithms, since it involves solving a system of linear equations at each iteration.

3.5 Issues with Newton's Method

However, Newton's method can be more computationally expensive than other algorithms, since it requires computing the Hessian matrix at each iteration. In addition, it can be unstable when the Hessian is poorly conditioned or when the objective function is non-convex. The main computational problem caused by Newton Method while calculating the Hessians paved a way to other class of methods called **Quasi-Newton** Methods which are discussed in the following sections.

4 Quasi-Newton Methods

We start by making a second-order approximation m_k of the objective function f around the point x_k as follows,

$$f(x_k + p_k) \approx m_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d \tag{3}$$

$$\therefore \nabla m_k(d) = \nabla f(x_k) + B_k d \tag{4}$$

We wish to minimize m_k with respect to d to obtain the search direction d_k. To do this, we set the first derivative of m_k to zero, which gives us,

$$\nabla f(x_k) + B_k d_k = 0$$
$$\therefore d_k = -B_k^{-1} \nabla f(x_k)$$

The update rule is given by,

$$x_{k+1} = x_k + \alpha_k d_k$$
$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$$
$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k) \tag{5}$$

The value of α_k is obtained by the Backtracking Line Search algorithm, which is described in section 4.1. In contrast to Newton's method, the B_k is calculated iteratively, avoiding the need to compute its value at every iteration. B_k essentially is an approximation of the Hessian matrix of f at x_k.

Given a function f, a starting point x_0, a tolerance $\epsilon > 0$ and inverse Hessian approximation H_0, the Quasi-Newton method is given by,

Algorithm 2 Quasi-Newton Method

$k \leftarrow 0$
while $\|\nabla f(x_k)\| > \epsilon$ **do**
  $p_k \leftarrow -H_k \nabla f_k$ or $-B_k^{-1} \nabla f_k$
  $\alpha_k \leftarrow$ Backtracking Line Search
  $x_{k+1} \leftarrow x_k + \alpha_k p_k$
  $s_k \leftarrow x_{k+1} - x_k$
  $y_k \leftarrow \nabla f(x_{k+1}) - \nabla f(x_k)$
  Calculate H_{k+1} or B_{k+1} using eq. 10
  $k \leftarrow k + 1$
end while
return x_k

4.1 Backtracking Line Search

The Backtracking Line Search algorithm is used to find the value of α_k in eq. (5). Such that they satisfy the sufficient decrease condition (6) and the curvature condition (7) which are given by,

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T d_k \tag{6}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f(x_k)^T d_k \tag{7}$$

where $0 < c_1 < c_2 < 1$ are constants. The Backtracking Line Search algorithm is given by,

Algorithm 3 Backtracking Line Search

Choose $\bar{\alpha} > 0$ and $\rho \in (0, 1)$
$\alpha \leftarrow \bar{\alpha}$
while $f(x_k + \alpha d_k) > f(x_k) + c_1 \alpha \nabla f(x_k)^T d_k$ **do**
  $\alpha \leftarrow \rho \alpha$
end while
return α

The algorithm is guarenteed to give a step size α_k that satisfies the sufficient decrease condition (6). This is because α_k will eventually become small enough to satisfy it. The obtained step size is also not too small, because it is within a factor of ρ of previous step size α_k/ρ, which did not satisfy the sufficient decrease condition.

4.2 Calculating B_k

We would like B_k to have the following properties,

1. It should be easy to compute $B_k^{-1} \nabla f(x_k)$.
2. m_{k+1} should match the curvature of f at x_k and x_{k+1}.

3. $\|B - B_k\|$ should be small.

To match the curvature of m_{k+1} with f at x_k we need,

$$\nabla m_{k+1}|_{d=0} = \nabla f(x_{k+1})$$

This is already true from eq. (9). As $x_{k+1} = x_k + \alpha_k d_k$ we have, $x_k = x_{k+1} - \alpha_k d_k$. Hence to match the curvature at x_{k+1} we need,

$$\begin{aligned}
\nabla m_{k+1}|_{d=-\alpha_k d_k} &= \nabla f(x_k) \\
\nabla f(x_{k+1}) - \alpha_k B_{k+1} d_k &= \nabla f(x_k) \\
\alpha_k B_{k+1} d_k &= \nabla f(x_{k+1}) - \nabla f(x_k) \\
B_{k+1}(x_{k+1} - x_k) &= \nabla f(x_{k+1}) - \nabla f(x_k) \\
B_{k+1} s_k &= y_k
\end{aligned}$$

This is known as the **Secant Equation**. Here,

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

As B_{k+1} is symmetric it has $n(n+1)/2$ degrees of freedom. The secant equation imposes n conditions, and the positive definiteness of B_{k+1} imposes an additional n conditions. Hence, these conditions are insufficient to determine B_{k+1} uniquely. Therefore, we try to find the B_{k+1} closest to B_k. That is,

$$\begin{aligned}
B_{k+1} &= \min_B \|B - B_k\|^2 \\
\text{subject to } & Bs_k = y_k, \quad B = B^T
\end{aligned} \tag{8}$$

The choice of norm affects the obtained solution, giving us different Quasi-Newton methods.

4.3 Davidon-Fletcher-Powell (DFP) Formula

A popular choice of norm is the weighted Frobenius norm which allows an easy solution to the minimization problem. The Frobenius norm is given by,

$$\|A\|_F^2 = \sum_{i,j} A_{ij}^2$$

The weighted Frobenius norm is given by,

$$\|A\|_W^2 = \left\| W^{1/2} A W^{1/2} \right\|_F^2$$

The weight matrix W can be any matrix that satisfies the relation $W y_k = s_k$. One such choice is $W = \bar{G_k}^{-1}$ where $\bar{G_k}$ is the average Hessian i.e.,

$$\bar{G_k} = \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau$$

$\bar{G_k}$ is a valid choice of W as it satisfies the relation $y_k = \bar{G_k} s_k$. It can be shown using the Taylor series expansion as follows,

$$\nabla f(x_k + \alpha_k p_k) = \nabla f(x_k) + \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) p_k d\tau$$
$$\nabla f(x_k + \alpha_k p_k) - \nabla f(x_k) = \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) p_k d\tau$$
$$y_k = \bar{G_k} \alpha_k p_k$$
$$y_k = \bar{G_k} s_k$$

With this choice of weight matrix and norm, the unique solution to the minimization problem is given by,

$$B_{k+1} = (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T$$

where

$$\rho_k = \frac{1}{y_k^T s_k}$$

We, however, require $B_k^{-1} = H_k$ to be easily computable for the update eq. (5). Hence we use the Sherman-Morrison-Woodbury formula to obtain the updated equation of H_k as follows,

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k} \tag{9}$$

4.4 Broyden-Fletcher-Goldfarb-Shanno (BFGS) Formula

If instead of imposing the conditions eq. (8) on B_{k+1} we impose similar conditions on H_{k+1}, we get the BFGS update formula. The corresponding conditions are,

$$H_{k+1}^2 = \min_H \|H - H_k\|^2$$
$$\text{subject to } Hy_k = s_k, \quad H = H^T$$

With the same choice of weight matrix and norm as before, the unique solution to the minimization problem is given by,

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \tag{10}$$

where

$$\rho_k = \frac{1}{y_k^T s_k}$$

5 Analysis

5.1 Convergence Analysis

5.1.1 Quadratic Convergence of Newton's Method:

Theorem: Suppose that f is twice differentiable and that the Hessian $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of a solution x^* at which the sufficient conditions are satisfied. Consider the iteration $x_{k+1} = x_k + p_k$, where p_k is given by $p_k^N = -\nabla^2 f_k^{-1} \nabla f_k$, then if the starting point x_0 is sufficiently close to x^*, then the sequence of iterates converges to x^* and the rate of convergence is quadratic.

Proof: From the definition of the Newton step and the optimality condition $\nabla f_* = 0$ we have that

$$\begin{aligned} x_{k+1} - x^* &= x_k + p_k^N - x^* \\ &= x_k - x^* - \nabla^2 f_k^{-1} \nabla f_k \\ &= \nabla^2 f_k^{-1} \left[\nabla^2 f_k (x_k - x^*) - (\nabla f_k - \nabla f_*) \right] \end{aligned}$$

From taylor's theorem, we can write:

$$\nabla f_k - \nabla f_* = \int_0^1 \nabla^2 f(x_k + t(x^* - x_k))(x_k - x^*) dt$$

So, we have,

$$\begin{aligned} &\|\nabla 2 f(x_k)(x_k - x^*) - (\nabla f_k - \nabla f(x^*))\| \\ &= \left\| \int_0^1 \left[\nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k)) \right] (x_k - x^*) dt \right\| \\ &\leq \int_0^1 \left\| \nabla^2 f(x_k) - \nabla^2 f(x_k + t(x^* - x_k)) \right\| \|x_k - x^*\| dt \\ &\leq \|x_k - x^*\|^2 \int_0^1 Lt \; dt = \frac{1}{2} L \|x_k - x^*\|^2 \end{aligned}$$

where L is the Lipschitz constant for $\nabla^2 f(x)$ for x near x^*. Since $\nabla^2 f(x^*)$ is nonsingular, there is a radius $r > 0$ such that $\left\| \nabla^2 f_k^{-1} \right\| \leq 2 \left\| \nabla^2 f(x^*)^{-1} \right\|$ for all x_k with $\|x_k - x^*\| \leq r$. By substituting the expression for the difference in x_{k+1} and x_k into the above equation, we get,

$$\left\|x_k + p_k^N - x^*\right\| \leq L\left\|\nabla^2 f(x^*)^{-1}\right\|\|x_k - x^*\|^2 = \tilde{L}\|x_k - x^*\|^2$$

where $\tilde{L} = L\left\|\nabla^2 f(x^*)^{-1}\right\|$. Choosing x_0 such that $\|x_0 - x^*\| = \min(r, 1/(2\tilde{L}))$, we can use this inequality inductively to deduce that the sequence converges to x^*, and the rate of convergence is quadratic.

5.1.2 Global Superlinear Convergence of BFGS

To prove the superlinear convergence of the BFGS method, we use the Dennis and More characterization of superlinear convergence. It applies to general nonlinear (not just convex) objective functions. We need the following assumption for the results that follow:

Assumption: The Hessian matrix G is Lipshitz continuous at x* that is,

$$\|G(x) - G(^*)\| \leq L\|x - x^*\|$$

for all x near x^*, where L is a positive constant.

We start by introducing:

$$\tilde{s_k} = G_*^{1/2} s_k, \quad \tilde{y_k} = G_*^{-1/2} y_k, \quad \tilde{B_k} = G_*^{-1/2} B_k G_*^{-1/2}$$

where $G_* = G(x^*)$ and x^* is a minimizer of f. We also define:

$$\cos\tilde{\theta_k} = \frac{\tilde{s_k^T}\tilde{B_k}\tilde{s_k}}{\|s_k\|\left\|\tilde{B_k}\tilde{s_k}\right\|} \qquad \tilde{q_k} = \frac{\tilde{s_k^T}\tilde{B_k}\tilde{s_k}}{\|s_k\|^2}$$

$$\tilde{M_k} = \frac{\|\tilde{y_k}\|^2}{\tilde{y_k^T}\tilde{s_k}} \qquad \tilde{m_k} = \frac{\tilde{y_k^T}\tilde{s_k}}{\tilde{s_k^T}\tilde{s_k}}$$

The BFGS update formula is as follows:

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

By premultiplying and postmultiplying the BFGS update formula by $G_*^{1/2}$ and grouping terms appropriately, we obtain

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

Since this expression has precisely the same form as the BFGS formula, it follows from the proof for convergence of BFGS that,

$$\psi(\tilde{B}_k) = \psi(\tilde{B}_k) + (\tilde{M}_k - \ln \tilde{m}_k - 1) + \left[1 - \frac{\tilde{q_k}}{\cos^2 \tilde{\theta_k}} + \ln \frac{\tilde{q_k}}{\cos^2 \tilde{\theta_k}}\right] + \ln cos^2 \tilde{\theta_k}$$

Now, since $y_k = \tilde{G_k} s_k$, we have,

$$y_k - G_* s_k = (\tilde{G_k} - G_*) s_k$$

and thus,

$$\tilde{y_k} - \tilde{s_k} = G_*^{-1/2} (\tilde{G_k} - G_*) G_*^{-1/2} \tilde{s_k}$$

By our Lipschitz continuity assumption for the Hessian G, we have,

$$\|\tilde{y_k} - \tilde{s_k}\| \leq \left\|\tilde{G_k}_*^{-1/2}\right\|^2 \|\tilde{s_k}\| \left\|\tilde{G_k} - G_*\right\| \leq \left\|G_*^{-1/2}\right\|^2 \|\tilde{s_k}\| L \epsilon_k$$

where ϵ_k is defined by

$$\epsilon_k = \max(\|x_{k+1} - x^*\|, \|x_k - x^*\|)$$

We have thus shown that

$$\frac{\|\tilde{y_k} - \tilde{s_k}\|}{\|\tilde{s_k}\|} \leq \tilde{c} \epsilon_k \tag{11}$$

for some positive constant $\tilde{c}$. This inequality plays an important role in superlinear convergence, as we will prove now:

Theorem: Suppose that f is twice continuously differentiable and that the iterates generated by the BFGS algorithm converge to a minimizer x at which the Lipschitz continuity of the Hessian holds. Suppose also that $\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty$ holds. Then x_k converges to x^* at a superlinear rate.

Proof: From the inequality $\frac{\|\tilde{y_k} - \tilde{s_k}\|}{\|\tilde{s_k}\|} \leq \tilde{c} \epsilon_k$, we have from the triangle inequality that,

$$\|\tilde{y_k}\| - \|\tilde{s_k}\| \leq \bar{c} \epsilon_k \|\tilde{s_k}\|, \qquad \|\tilde{s_k}\| - \|y_k\| \leq \bar{c} \epsilon_k \|\tilde{s_k}\|$$

so that,

$$(1 - \bar{c} \epsilon_k) \|\tilde{s_k}\| \leq \|\tilde{y_k}\| \leq (1 + \bar{c} \epsilon_k) \|\tilde{s_k}\| \tag{12}$$

By squaring 11 and using 12, we obtain,

$$2\tilde{y_k^T} \geq (1 - 2\bar{c}\epsilon_k + 1 - \bar{c}^2\epsilon_k^2)\|\tilde{s_k}\|^2 = 2(1 - \bar{c}\epsilon_k)\|\tilde{s_k}\|^2$$

It follows from the definition of $\tilde{m_k}$ that,

$$\tilde{m_k} = \frac{\left\|\tilde{y_k}^T\right\|\tilde{s_k}}{\|\tilde{s_k}\|^2} \geq 1 - \bar{c}\epsilon_k \tag{13}$$

By combining 12 and 13, we obtain also that,

$$\tilde{M_k} = \frac{\|\tilde{y_k}\|^2}{\tilde{y_k^T}\tilde{s_k}} \geq 1 - \bar{c}\epsilon_k \tag{14}$$

Since $x_k \to x^*$, we have that $\epsilon_k \to 0$ and thus by 14 there exists a positive constant $c > \bar{c}$ such that the following inequalities hold for all sufficiently large k:

$$\tilde{M_k} \leq 1 + \frac{2\bar{c}}{1 - \bar{c}\epsilon_k}\epsilon_k \leq 1 + c\epsilon_k \tag{15}$$

We make use of the nonpositiveness of the function $h(t) = 1 - t + \ln t$. Therefore, we have,

$$\frac{-x}{1-x} - \ln(1-x) = h\left(\frac{1}{1-x}\right) \leq 0$$

Now, for k large enough we can assume that $\bar{c}\epsilon_k < \frac{1}{2}$ and therefore,

This relation and 13 that for sufficiently large k, we have

$$\sum_{j=0}^{\infty}\left(\ln\frac{1}{\cos^2\tilde{\theta_j}} - \left[1 - \frac{\tilde{q_j}}{\cos^2\tilde{\theta_j}} + \ln\frac{\tilde{q_j}}{\cos^2\tilde{\theta_j}}\right]\right) \leq \psi(\tilde{B_0}) + 3c\sum_{j=0}^{\infty}\epsilon_j < \infty$$

Since the term in the square brackets in nonpositive, and since $\ln\left(1/\cos^2\tilde{\theta_j}\right) \geq 0$ for all j, we obtain the two limits:

$$\lim_{j\to\infty}\cos\tilde{\theta_j} = 1 \qquad \lim_{j\to\infty}\tilde{q_j} = 1 \tag{16}$$

The essence of the result has now been proven; we need only to interpret these limits in terms of the Dennis–Moré characterization of superlinear convergence.

Note that,

$$\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} = \frac{\|B_k s_k\|^2}{(s_k^T B_k s_k)^2}\frac{s_k^T B_k s_k}{\|s_k\|^2} = \frac{2}{\cos^2\theta_k}$$

Thus we have,

$$\frac{\left\|G_*^{-1/2}(B_, - G_*)s_k\right\|^2}{\left\|G_*^{1/2}s_k\right\|^2} = \frac{\left\|(\tilde{B_k} - I)\tilde{s_k}\right\|}{\|\tilde{s_k}\|^2}$$

$$= \frac{\left\|\tilde{B_k}\tilde{s_k}\right\| - 2\tilde{s_k^T}\tilde{B_k}\tilde{s_k} + \tilde{s_k^T}\tilde{s_k}}{\tilde{s_k^T}\tilde{s_k}}$$

$$= \frac{\tilde{q_k^2}}{\cos\tilde{\theta_k^2}} - 2\tilde{q_k} + 1$$

Since by 16 the right-hand-side converges to 0, we conclude that,

$$\lim_{k\longrightarrow\infty} \frac{\|(B_k - G_*)s_k\|}{\|s_k\|} = 0$$

Hence the unit step length $\alpha_k = 1$ will satisfy the Wolfe conditions near the solution, and hence that the rate of convergence is superlinear.

5.2 Time Complexity

5.2.1 Newton's Method

The time complexity of Newton's method is $O(n^3)$ for each iteration, excluding the cost of function and gradient evaluation. This is because we are solving a linear system of equations at each iteration, requiring the inversion of the Hessian matrix, which is $O(n^3)$.

5.2.2 Quasi-Newton Methods

The time complexity of the Quasi-Newton method is only $O(n^2)$ for each iteration, excluding the cost of function and gradient evaluation. This is because the algorithm 2 including the update step eq. (9) or eq. (10) do not require any linear system solves or matrix-matrix multiplications. The only operation that is required is the calculation of the inverse Hessian approximation, which is $O(n^2)$. In addition unlike Newton's method, the Hessian matrix is not required to be computed at every iteration.

6 Experiments and Observations

We tried running Gradient Descent, Newton's method and BFGS over the following three functions. The stopping criteria used is: $\|\nabla f\| < \epsilon$, where, $\epsilon = 0.01$. Newton's method reaches the minima in least number of iterations, since it uses the exact hessian; though the time required per iteration is high due to hessian calculation at each step. BFGS needs less iterations as well, but more than

Newton's method. Gradient descent with a fixed step size requires the highest number of iterations and takes least amount of time per iteration. The step size for Newton's Method is 1, for gradient descent is 0.1 or 0.00001 (depending on the function) and for BFGS is dynamically calculated using Backtracking and Wolfe conditions. Following are the contour plots of the functions considered, and the (x, y) values at each iterate of the three algorithms:

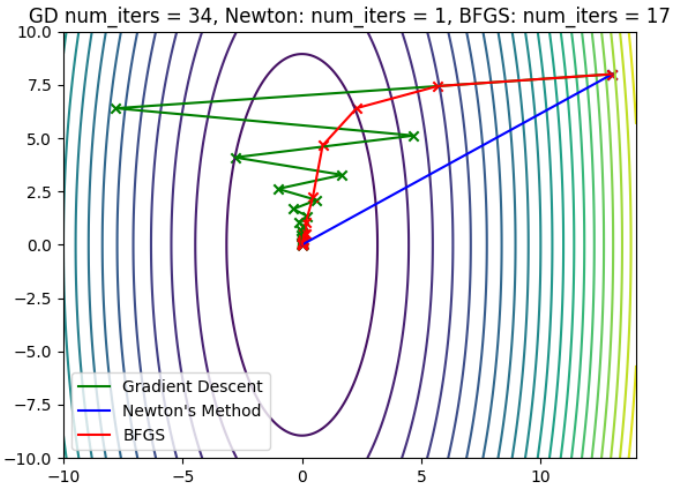

Figure 1: Contour Plot

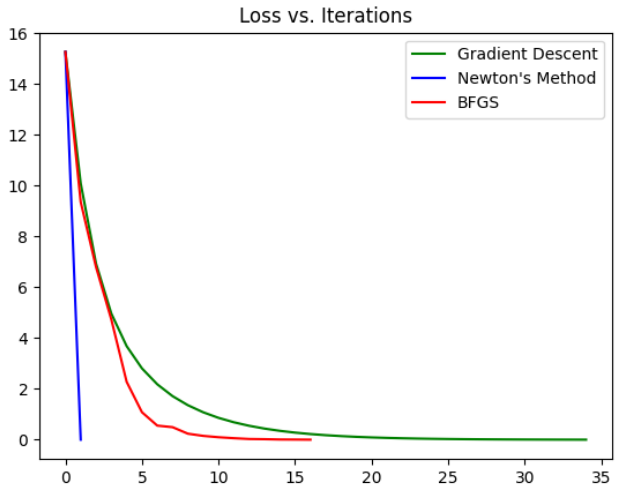

Figure 2: Loss

- Function: $f(x, y) = 8x^2 + y^2$
- Initial Point: $(x_0, y_0) = (13, 8)$
- Gradient Descent Step Size: 0.1

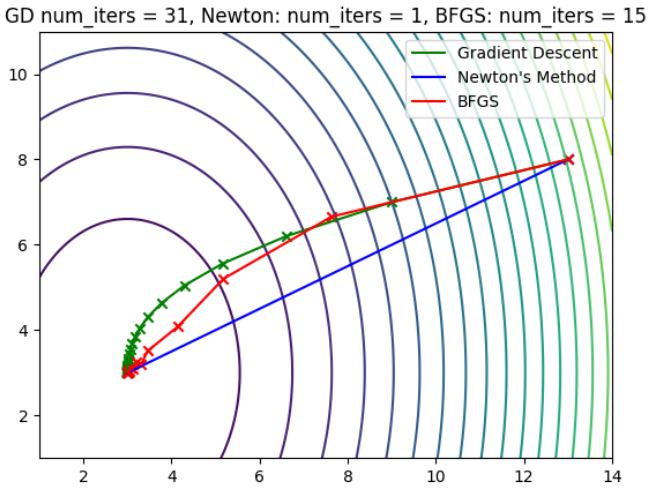

Figure 3: Contour Plot

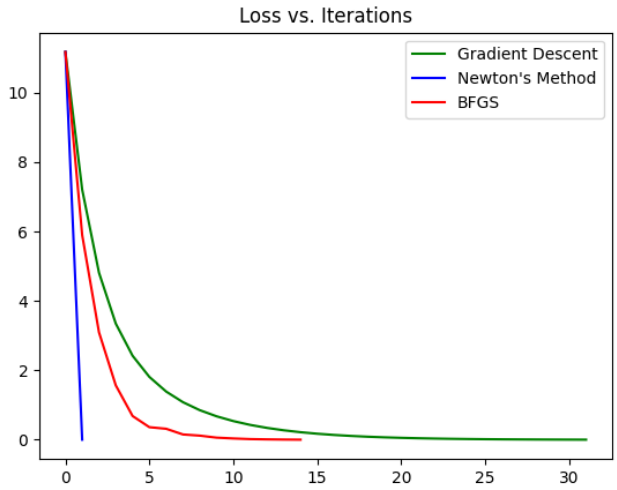

Figure 4: Loss

- Function: $f(x, y) = (x - 2)^2 + (x - 4)^2 + (y - 3)^2$
- Initial Point: $(x_0, y_0) = (13, 8)$
- Gradient Descent Step Size: 0.1

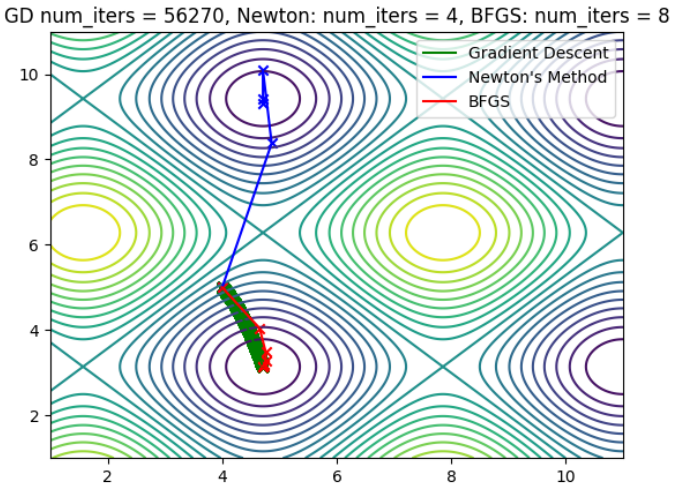

Figure 5: Contour Plot

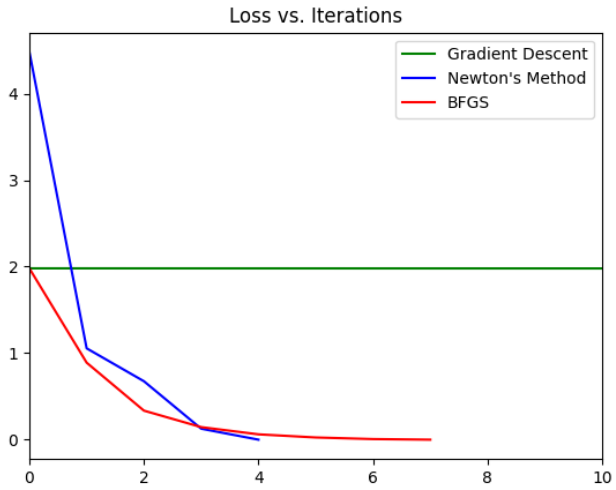

Figure 6: Loss

- Function: $f(x, y) = \sin x + \cos y$
- Initial Point: $(x_0, y_0) = (4, 5)$
- Gradient Descent Step Size: 0.0001

References

[1] Constantine Caramanis. *Quasi Newton and BFGS*. Nov 2020.

[2] R. Fletcher. *Practical Methods of Optimization*. A Wiley-Interscience publication. Wiley, 2000.

[3] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2000.

[4] Wikipedia contributors. Sherman–morrison formula — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Sherman%E2%80%93Morrison_formula&oldid=1126413915`, 2022. [Online; accessed 10-April-2023].

[5] Wikipedia contributors. Broyden–fletcher–goldfarb–shanno algorithm — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Broyden%E2%80%93Fletcher%E2%80%93Goldfarb%E2%80%93Shanno_algorithm&oldid=1145392248`, 2023. [Online; accessed 10-April-2023].