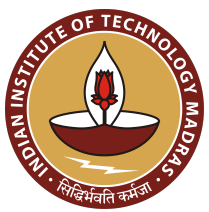
# Quasi-Newton Methods

Saurav Samantaray

Department of Mathematics

Indian Institute of Technology Madras

March 4, 2024



# Quasi-Newton Methods

- Quasi-Newton methods (like steepest descent), require only the gradient of the objective function to be supplied at each iterate.
- By measuring the changes in gradients, they construct a model of the objective function that is good enough to produce super-linear convergence.
- The improvement over steepest descent is dramatic, especially on difficult problems.
- Moreover, since second derivatives are not required (unlike Newton's), quasi-Newton methods are sometimes more efficient than Newton's method.
- Today, optimization software libraries contain a variety of quasi-Newton algorithms for solving unconstrained, constrained, and large-scale optimization problems.

# THE BFGS METHOD

- BFGS method, named for its discoverers Broyden, Fletcher, Goldfarb, and Shanno.
- Consider the following quadratic model of the objective function at the current iterate $x_k$ :

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p. \tag{1}$$

$B_k$ is a $n \times n$ symmetric positive definite matrix that will be revised or updated at every iteration.

- As the model is first-order accurate the function value $f_k$ and the gradient $\nabla f_k$, both match at $p = 0$.
- The minimiser of this model is

$$p_k = -B_k^{-1} \nabla f_k, \tag{2}$$

# THE BFGS METHOD

- $p_k$ is used as the search direction, and the new iterate is

$$x_{k+1} = x_k + \alpha_k p_k \tag{3}$$

  where the step length $\alpha_k$ is chosen to satisfy the Wolfe conditions.
- Instead of computing $B_k$ afresh at every iteration, Davidon proposed to update it in a simple manner to account for the curvature measured during the (most recent) previous step.
- Suppose that we have generated a new iterate $x_{k+1}$ and wish to construct a new quadratic model, of the form

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p. \tag{4}$$

- What requirements should we impose on $B_{k+1}$, based on the knowledge gained during the latest step?

# THE BFGS METHOD

- One reasonable requirement is that the gradient of $m_{k+1}$ should match the gradient of the objective function $f$ at the latest two iterates $x_k$ and $x_{k+1}$, i.e.

$$\nabla m_{k+1}(0) = \nabla f_{k+1} \quad \text{(condition at the second point is satisfied)}$$

- The first condition can be written mathematically as

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k$$

- By rearranging we get

$$B_{k+1}\alpha_k p_k = \nabla f_{k+1} - \nabla f_k \tag{5}$$

# THE BFGS METHOD

- Introduce the notations

$$s_k = x_{k+1} - x_k = \alpha_k p_k$$
$$y_k = \nabla f_{k+1} - \nabla f_k$$

- With the above notation (5) becomes

$$B_{k+1} s_k = y_k \tag{6}$$

  which is called the secant equation.
- Given the displacements $s_k$ and the change of gradients $y_k$,the secant equation requires that the symmetric positive definite matrix $B_{k+1}$ map $s_k$ into $y_k$.

# THE BFGS METHOD

- This will be possible only if $s_k$ and $y_k$ satisfy the curvature condition

$$s_k^T y_k > 0. \tag{7}$$

- To get the above condition pre-multiply (6) with $s_k^T$.
- When $f$ is strongly convex,the inequality (7) will be satisfied for any two points $x_k$ and $x_{k+1}$.
- However, this condition will not always hold automatically for non-convex functions.
- The curvature condition (7) has to explicitly enforced, by imposing restrictions on the line search procedure that chooses the step length $\alpha$, in such cases.

# THE BFGS METHOD

- Consider the second Wolfe condition:

$$\begin{aligned} & \nabla f_{k+1}^T p_k \geq c_2 \nabla f_k^T p_k \\ \Longrightarrow & \nabla f_{k+1}^T s_k \geq c_2 \nabla f_k^T s_k \\ \Longrightarrow & y_k^T s_k \geq (c_2 - 1)\alpha_k \nabla f_k^T p_k \end{aligned} \tag{8}$$

  $c_2 < 1$ and $p_k$ is a descent direction the term on the right is positive, and the curvature condition (7) holds.

- When the curvature condition is satisfied, the secant equation (6) always has a solution $B_{k+1}$.
- In fact, it admits an infinite number of solutions, since the $n(n+1)/2$ degrees of freedom in a symmetric positive definite matrix exceed the $n$ conditions imposed by the secant equation.

# THE BFGS METHOD

## Definition

A minor of $B$ of order $k$ is principal if it is obtained by deleting $n-k$ rows and $n-k$ columns columns with the same numbers. The leading principal minor of $B$ of order $k$ is the minor of order $k$ obtained by deleting the last $n-k$ rows and columns. We write $D_k$ for the leading principal minor of order $k$

## Theorem

Let $B$ be a symmetric $n \times n$ matrix. Then we have:
$B$ is positive definite iff $D_k > 0$ for all leading principal minors.

- The requirement of positive definiteness imposes $n$ additional inequalities —all leading principal minors must be positive—but these conditions do not absorb the remaining degrees of freedom.

# THE BFGS METHOD

- To determine $B_{k+1}$ uniquely, we impose the additional condition that among all symmetric matrices satisfying the secant equation, $B_{k+1}$ is, in some sense, closest to the current matrix $B_k$.

$$\begin{aligned} &\min_B ||B - B_k|| \\ &\text{subject to } B = B^T, \quad Bs_k = y_k \end{aligned} \tag{9}$$

  where $s_k$ and $y_k$ satisfy the curvature condition (7) and $B_k$ is symmetric and positive definite.
- BFGS updating can be derived by making a simple change in the argument.
- Instead of imposing conditions on the Hessian approximations $B_k$ , we impose similar conditions on their inverses $H_k$

# THE BFGS METHOD

- The updated approximation $H_{k+1}$ must be symmetric and positive definite, and must satisfy the secant equation (6), now written as

$$H_{k+1}y_k = s_k$$

- The condition of closeness to $H_k$ is now specified by the following analogue

$$\begin{aligned} &\min_H ||H - H_k|| \\ &\text{subject to } H = H^T, \quad Hy_k = s_k \end{aligned} \tag{10}$$

- The norm is a weighted Frobenius norm given by:

$$||A||_W = ||W^{1/2}AW^{1/2}||_F$$

where $||.||_F$ is defined by $||C||_F^2 = \sum_{i=1}^n \sum_{i=1}^n c_{ij}^2$

# THE BFGS METHOD

- The weight matrix $W$ is any matrix satisfying $Ws_k = y_k$.
- The unique solution $H_{k+1}$ is given by

$$\text{BFGS} \qquad H_{k+1} = (I - \rho_k s_K y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \quad (11)$$

- where $\rho_k = \frac{1}{y_k^T s_k}$
- Just one issue has to be resolved before we can define a complete BFGS algorithm:
- How should we choose the initial approximation $H_0$ ?
- Unfortunately, there is no magic formula that works well in all cases.
- We can use specific information about the problem, for instance by setting it to the inverse of an approximate Hessian calculated by finite differences at $x_0$.
- Worst case, we can simply set it to be the identity matrix, or a multiple of the identity matrix, where the multiple is chosen to reflect the scaling of the variables.

# THE BFGS METHOD

**Algorithm 6.1** (BFGS Method).

Given starting point $x_0$, convergence tolerance $\epsilon > 0$,
inverse Hessian approximation $H_0$;
$k \leftarrow 0$;
**while** $\|\nabla f_k\| > \epsilon$;
Compute search direction

$$p_k = -H_k \nabla f_k;$$

Set $x_{k+1} = x_k + \alpha_k p_k$ where $\alpha_k$ is computed from a line search
procedure to satisfy the Wolfe conditions (3.6);
Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$;
Compute $H_{k+1}$ by means of (6.17);
$k \leftarrow k + 1$;
**end (while)**