QUASI-NEWTON METHODS

David F. Gleich

April 3, 2023

The material here is from Chapter 6 in Nocedal and Wright, and Section 12.3 in Griva, Sofer, and Nash.

The idea behind Quasi-Newton methods is to make an optimization algorithm *with only a function value and gradient* converge more quickly than steepest descent. That is, a Quasi-Newton method does not require a means to evaluate the Hessian matrix at the current iterate, as in a Newton method. Instead, the algorithm constructs a matrix that resembles the Hessian as it proceeds.

In fact, there are many ways of doing this, and so there is really a family of Quasi-Newton methods.

1 QUASI-NEWTON IN ONE VARIABLE: THE SECANT METHOD

In a one dimensional problem, approximating the Hessian simplifies to approximating the second derivative: $f''(x) \approx \frac{f'(x+h)-f'(x)}{h}$. Thus, the fact that this is possible is not unreasonable. Using a related approximation in a one-dimensional optimization algorithm results in a procedure called the *Secant method*:

$$\underbrace{\text{“}x_{k+1} = x_k - \frac{1}{f''(x_k)} f'(x_k)\text{”}}_{\text{One dimensional Newton}} \quad \to \quad x_{k+1} = x_k - \underbrace{\frac{(x_k - x_{k-1})}{f'(x_k) - f'(x_{k-1})}}_{\approx 1/f''(x_k)} f'(x_k)$$

This new update is trying to approximate the Newton update by approximating the second derivative information.

The secant method converges superlinearly, under appropriate conditions; so this idea checks out in one-dimension.

2 QUASI-NEWTON IN GENERAL

Quasi-Newton methods are line-search methods that compute the search direction by trying to approximate the Newton direction:

$$\text{“}\boldsymbol{H}(\mathbf{x}_k)\mathbf{p} = -\mathbf{g}\text{”}$$

without using the matrix $\boldsymbol{H}(\mathbf{x}_k)$. They work by computing

$$\boldsymbol{B}_k \quad \text{“that behaves like”} \quad \boldsymbol{H}(\mathbf{x}_k).$$

Once we compute $\mathbf{x}_{k+1}$, then we update $\boldsymbol{B}_k \to \boldsymbol{B}_{k+1}$. Thus, a Quasi-Newton method has the general iteration:

```
initialize B_0, and k = 0
for k = 0,... and while x_k does not satisfy the conditions we want ...
  solve for the search direction B_k p_k = -g
  compute a line search α_k
  update x_{k+1} = x_k + α p_k
  update B_{k+1} based on x_{k+1}
```

We can derive different Quasi-Newton methods by changing how we update $\boldsymbol{B}_{k+1}$ from $\boldsymbol{B}_k$.

3 THE SECANT CONDITION

While there are many ways of updating $\boldsymbol{B}_{k+1}$ from $\boldsymbol{B}_k$, a random choice is unlikely to provide any benefit, and may making things considerably worse. Thus, we want to start from a principled approach.

Recall that the Newton direction $\boldsymbol{H}_k \mathbf{p}_k = -\mathbf{g}$ arises as the unconstrained minimizer of

$$m_k^N(\mathbf{p}) = f_k + \mathbf{g}_k^T \mathbf{p} + \tfrac{1}{2}\mathbf{p}^T \boldsymbol{H}_k \mathbf{p}$$

when $\boldsymbol{H}_k$ is positive definite.

The model for Quasi-Newton methods uses $\boldsymbol{B}_k$ instead of $\boldsymbol{H}_k$:

$$m_k^Q(\mathbf{p}) = f_k + \mathbf{g}_k^T \mathbf{p} + \tfrac{1}{2}\mathbf{p}^T \boldsymbol{B}_k \mathbf{p}$$

so one common requirement for $\boldsymbol{B}_k$ is that it remains positive definite. This requirement is relaxed for some Quasi-Newton methods.

However, all Quasi-Newton methods require:

$$\nabla m_{k+1}^Q(0) = \mathbf{g}(\mathbf{x}_{k+1})$$

and

$$\nabla m_{k+1}^Q(-\alpha_k \mathbf{p}_k) = \mathbf{g}(\mathbf{x}_k).$$

In other works, a Quasi-Newton method has the property that the gradient of the model function $m_{k+1}^Q(\mathbf{p})$ *has the same gradient* as f at $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$.

This requirement imposes some conditions on $\boldsymbol{B}_{k+1}$:

$$\nabla m_{k+1}^Q(-\alpha \mathbf{p}_k) = \mathbf{g}(\mathbf{x}_{k+1}) - \alpha_k \boldsymbol{B}_{k+1}\mathbf{p}_k = \mathbf{g}(\mathbf{x}_k) \quad \longrightarrow \quad \boldsymbol{B}_{k+1}\alpha_k \mathbf{p}_k = \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k).$$

Note that $\alpha_k \mathbf{p}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. If we define:

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k \quad \text{and} \quad \mathbf{y}_k = \mathbf{g}(\mathbf{x}_{k+1}) - \mathbf{g}(\mathbf{x}_k).$$

Then Quasi-Newton methods require:

$$\boldsymbol{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k,$$

which is called the *secant condition*.

If we write this out for a one-dimensional problem:

$$b_{k+1}(x_{k+1} - x_k) = f'(x_{k+1}) - f'(x_k).$$

This equation is identical to the approximation of $f''(x_k)$ used in the secant method.

Quiz Is it always possible to find such a $\boldsymbol{B}_{k+1}$? Suppose that $\boldsymbol{B}_k$ is symmetric, positive definite. Show that we need $\mathbf{y}_k^T \mathbf{s}_k > 0$ in order for $\boldsymbol{B}_{k+1}$ to be positive definite. If $\boldsymbol{B}_k = 1$ for a one dimensional problem, find a function where this isn't true.

3.1 THE SECANT CONDITION & STRONG WOLFE

If the line search routine guarantees the Strong Wolfe conditions, then the update will automatically satisfy the Secant condition.

Consider the Strong Wolfe conditions:

$$f(\mathbf{x}_k + \alpha \mathbf{p}_k) \le f(\mathbf{x}_k) + \alpha c_1 \mathbf{p}_k^T \mathbf{g}_k$$
$$|\mathbf{g}(\mathbf{x}_k + \alpha \mathbf{p}_k)^T \mathbf{p}| \le c_2 \mathbf{g}_k^T \mathbf{p}_k$$

The second condition implies:

$$\mathbf{g}_{k+1}^T \mathbf{p}_k \ge c_2 \mathbf{g}_k^T \mathbf{p}_k.$$

Now consider $\mathbf{y}_k^T \mathbf{s}_k$ for a step that satisfies Strong Wolfe:

$$\mathbf{y}_k^T \mathbf{s}_k = (\mathbf{g}_{k+1} - \mathbf{g}_k)^T (\alpha \mathbf{p}_k) = \alpha \mathbf{g}_{k+1}^T \mathbf{p}_k - \alpha \mathbf{g}_k^T \mathbf{p}_k \ge \alpha (c_2 - 1)\mathbf{g}_k^T \mathbf{p}_k.$$

But, we know $0 < c_2 < 1$, so $c_2 - 1 < 0$, and $\mathbf{g}_k^T \mathbf{p}_k < 0$. Thus, we satisfy the Secant condition!

4 FINDING THE UPDATE

We are getting closer to figuring out how to find such an update. There are many ways to derive the following updates, I'll just list them and state their properties.

4.1 DAVIDSON, FLETCHER, POWELL (DFP)

Let $\rho = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k}$.

$$\boldsymbol{B}_{k+1} = (\boldsymbol{I} - \rho_k \mathbf{y}\mathbf{s}^T)\boldsymbol{B}_k(\boldsymbol{I} - \rho_k \mathbf{s}\mathbf{y}^T) + \rho_k \mathbf{y}_k \mathbf{y}_k^T.$$

Clearly this matrix is symmetric when $\boldsymbol{B}_k$ is. Also, $\boldsymbol{B}_{k+1}$ is positive definite.

Quiz Show that $\boldsymbol{B}_{k+1}$ is positive definite.

This choice of $\boldsymbol{B}_{k+1}$ has the following optimality property:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{B} - \boldsymbol{B}_k\|_W \\ \text{subject to} & \boldsymbol{B}^T = \boldsymbol{B}, \boldsymbol{B}\mathbf{s}_k = \mathbf{y}_k \end{array}$$

where $\boldsymbol{W}$ is a weight based on the average Hessian. (This gives a "closest" matrix view on the solution.

4.2 BROYDEN, FLETCHER, GOLDFARB, SHANNO (BFGS) – "STANDARD"

Because we compute the search direction by solving a system with the approximate Hessian matrix:

$$\boldsymbol{B}_k \mathbf{p}_k = -\mathbf{g}_k,$$

the BFGS update constructs an approximation of the *inverse Hessian* instead. Suppose that

$$\boldsymbol{T}_k \text{ "behaves like" } \boldsymbol{H}(\mathbf{x})^{-1}.$$

Then

$$\boldsymbol{T}_{k+1}\mathbf{y}_k = \mathbf{s}_k$$

is the secant condition for the inverse. This helps because now we can find search directions via

$$\mathbf{p}_k = -\boldsymbol{T}_k \mathbf{g}_k,$$

via a matrix-vector multiplication instead of a linear solve.

The BFGS method uses the update:

$$\boldsymbol{T}_{k+1} = (\boldsymbol{I} - \rho_k \mathbf{s}\mathbf{y}^T)\boldsymbol{T}_k(\boldsymbol{I} - \rho_k \mathbf{y}\mathbf{s}^T) + \rho \mathbf{s}_k \mathbf{s}_k^T.$$

By the same proof, this update is also positive definite.

This choice has the following optimality property:

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{T} - \boldsymbol{T}_k\|_W \\ \text{subject to} & \boldsymbol{T}^T = \boldsymbol{T}, \boldsymbol{T}\mathbf{y}_k = \mathbf{s}_k \end{array}$$

where $\boldsymbol{W}$ is a weight based on the average Hessian.

4.3 SYMMETRIC RANK-1 (SR1) – FOR TRUST REGION METHODS

Both of the previous updates were rank-2 changes to $\boldsymbol{B}_k$ (or $\boldsymbol{T}_k$). The SR1 method is a rank-1 update to $\boldsymbol{B}_k$. Unfortunately, this update will not preserve positive definiteness. Nonetheless, it's frequently used in practice and is a reasonable choice for Trust Region methods that don't require a positive definite approximate Hessian.

Any rank-1 symmetric matrix is:

$$\sigma \mathbf{v}\mathbf{v}^T$$

and so the update is:

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \sigma \mathbf{v}\mathbf{v}^T.$$

Applying the Secant equation constrains $\mathbf{v}$, and we have:

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \frac{(\mathbf{y}_k - \boldsymbol{B}_k\mathbf{s}_k)(\mathbf{y}_k - \boldsymbol{B}_k\mathbf{s}_k)^T}{(\mathbf{y}_k - \boldsymbol{B}_k\mathbf{s}_k)^T\mathbf{s}_k}$$

or

$$\boldsymbol{T}_{k+1} = \boldsymbol{T}_k + \frac{(\mathbf{s}_k - \boldsymbol{T}_k\mathbf{y}_k)(\mathbf{s}_k - \boldsymbol{T}_k\mathbf{y}_k)^T}{(\mathbf{s}_k - \boldsymbol{T}_k\mathbf{y}_k)^T\mathbf{y}_k}.$$

The SR1 method tends to generate better approximations to the true Hessian than the other methods. For instance, if the search directions $\mathbf{p}_k$ are all linearly independent for $k = 1, \ldots, n$, and $f(\mathbf{x})$ is a simple quadratic model, then $\boldsymbol{T}_n$ is the inverse of the true Hessian.

4.4 BROYDEN CLASS

The Broyden class is a linear combination of the BFGS and the DFP method:

$$\boldsymbol{B}_{k+1} = (1 - \phi)\boldsymbol{B}_{k+1}^{\mathrm{BFGS}} + \phi\boldsymbol{B}_{k+1}^{\mathrm{DFP}}.$$

(This form requires the BFGS update for $\boldsymbol{B}$ and not $\boldsymbol{T}$.)

There are all sorts of great properties of the Broyden class, e.g. for the right choice of parameters, it'll reproduce the CG method.

5 PROPERTIES OF QUASI-NEWTON

Quasi-Newton has all sorts of great properties. Let's highlight a few of the theorems from Nocedal and Wright.

5.1 QUASI-NEWTON AND THE SHERMAN-MORRISON-WOODBURY FORMULA

We can convert quasi-Newton updates from updates to the approximate Hessian $\boldsymbol{B}_k$ to the approximate inverse Hessian $\boldsymbol{T}_k$ using the Sherman-Morrison-Woodbury formula. e.g. the BFGS update to the approximate Hessian is:

$$\boldsymbol{B}_{k+} = \boldsymbol{B} - \frac{1}{\mathbf{s}^T\boldsymbol{B}\mathbf{s}}\boldsymbol{B}_k\mathbf{s}\mathbf{s}^T\boldsymbol{B} + \frac{1}{\mathbf{y}^T\mathbf{s}}\mathbf{y}\mathbf{y}^T.$$

5.2 CONVERGENCE TO THE HESSIAN

Theorem (6.3). If $\mathbf{f}$ is a strongly convex quadratic,

$$\mathbf{f}(\mathbf{x}) = \mathbf{x}^T\mathbf{b} + \tfrac{1}{2}\mathbf{x}^T\boldsymbol{Q}\mathbf{x},$$

where $\boldsymbol{Q}$ is positive definite. Then the eigenvalues of

$$\boldsymbol{Q}^{1/2}\boldsymbol{B}_k\boldsymbol{Q}^{1/2}$$

monotonically converge to 1 if $\boldsymbol{B}_k$ is updated using a restricted Broyden class update.

5.3 COMPUTATION OF THE HESSIAN

Theorem (6.4). If $\mathbf{f}$ is a strongly convex quadratic,

$$\mathbf{f}(\mathbf{x}) = \mathbf{x}^T\mathbf{b} + \tfrac{1}{2}\mathbf{x}^T\boldsymbol{Q}\mathbf{x},$$

where $\boldsymbol{Q}$ is positive definite. If we use exact line search with an update from the Broyden class, then after n steps, starting from $\boldsymbol{B}_0 = \boldsymbol{I}$, we have that $\boldsymbol{B}_n = \boldsymbol{Q}$. (i.e. we recover the exact Hessian!)