Lecture 15: Quasi-Newton

Sondre Pedersen

March 7, 2025

1 Intro

Quasi Newton efficiently produce good search directions. It needs fewer iterations than Steepest descent, and is cheaper than Newton. This is the case because we use some cool techniques to approximate the Hessian.

Conditions for a good step length: Wolfe Conditions

- $f(x_k + \alpha p_k) \leq f(x_k) + c_1 + \alpha_k \nabla f_k^\top p_k$ Sufficient decrease (Armijo condition)
- $\nabla f(x_k + \alpha p_k)^\top p_k \geq c_2 \nabla f_k^\top p_k$ Desired slope (Curvature condition)

Newton's method - recap

Approximate ('model') $f(x)$ at x_k.

$$\begin{aligned} f(x_k + p) &\approx m_k(p) \\ &= f_k + \nabla f_k^\top p + \frac{1}{2} p^\top \nabla^2 f_k p \end{aligned}$$

The Newton direction: $p = arg\min_p m_k(p)$. If we assume that $\nabla^2 f_k > 0 \implies m_k(p)$ convex. This implies $\nabla m_k(p) = 0$ necessary and sufficient condition for min. $\nabla m_k(p) = \nabla f_k + \nabla^2 f_k p = 0 \implies p = -[\nabla^2 f_k]^{-1} \nabla f_k$. The matrix is invertible because we assumed $\nabla^2 f_k > 0$.

Hessian modification -
For $p_k = -B_k^{-1} \nabla f(x_k)$ to be a descent direction, we need $B_k > 0$. In general, this does not hold true for Newton, $B_k = \nabla^2 f(x_k)$. We therefore modify the Hession when it is not positive definite. This is done with algorithm 3.2. The problem with this method is that it uses expensive calculations.

2 Newton vs. Quasi-Newton

Newton's method -

$x_{k+1} = x_k + \alpha_k p_k \qquad p_k = -[\nabla^2 f_k]^{-1} \nabla f_k.$

- Advantage: Fast convergence (few iterations).
- Drawback: Expensive (at least for problems with many variables). Why is it expensive?
  - Calculating (and storing) the Hessian ($\nabla^2 f_k$).
  - Solve $\nabla^2 f_k p_k = -\nabla f_k$.

Quasi-Newton -

Makes an approximation of $f(x)$ at x_k with

$$f(x_k + p) \approx f_k + \nabla f_k^\top p + \frac{1}{2} p^\top B_k p.$$

The difference between this and Newton is the therm B_k instead of the Hessian. So the name of the game is to pick a good B_k.

We want

- $B_k > 0$ to ensure descent direction
- $B_k \approx \nabla^2 f_k$ to ensure fast convergence.
- Cheap computation, only using the gradient.

3 Secant condition

Quasi-Newton was invented by Bill Davidon around the mid 1950s. Came up with 'The Davidon-Fletcher-Powell' (DFP) update formula. The key in this update rule was the secant condition.

Consider

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^\top p + \frac{1}{2} p^\top B_k p$$
$$\nabla m_{k+1}(p) = \nabla f_{k+1} + B_k p$$

Now using the path (αp_k) between x_k and x_{k+1} to evaluate the gradient at x_{k+1}:

- $\nabla m_{k+1}(0) = \nabla f_{k+1}$
- $\nabla m_{k+1}(-\alpha p_k) = \nabla f_{k+1} - \alpha B_{k+1} p_k$

We want the second equation to equal ∇f_k. To achieve this, $B_{k+1}\alpha_k p_k = \nabla f_{k+1} - \nabla f_k$. Defining some variables $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$, the secant condition can be written as $B_{k+1} s_k = y_k$.

Same comes from Taylor expansion of $\nabla f(x_k)$.

$$\nabla f_{k+1} = \nabla f_k + \nabla^2 f_k (x_{k+1} - x_k) + \dots$$

That is, the secant condition implies $B_{k+1} \approx \nabla^2 f(x_k)$. But it does not tell us how to compute B_{k+1}. We can use some of the requirements from earlier to do this.

Positive definite requirement -

We want $B_{k+1} > 0$. Note that $s_k^\top y_k = s_k^\top B_{k+1} s_k$. So we must require that

$$s_k^\top y_k = (x_{k+1} - x_k)^\top (\nabla f_{k+1} - \nabla f_k) > 0.$$

This holds if the step length that we choose fulfills Wolfe condition. It also holds for any α if $f(x)$ is convex.

4 DFP update formula

Observation: There are infinitely many $B_{k+1} > 0$ that fulfills $B_{k+1} s_k = y_k$. So we choose B_{k+1} closest to B_k.

$$B_{k+1} = \arg\min_B ||B - B_k|| \qquad \text{s.t.} \qquad B = B^\top,\ B s_k = y_k$$

The norm you choose actually determines which Quasi-Newton method you get. The norm used in DFP is 'weighted Frobenius norm':

$$\begin{aligned} B_{k+1} =& (I - \rho_k y_k s_k^\top) B_k (I - \rho_k s_k y_k^\top) + \rho_k y_k y_k^\top \\ & \text{where} \\ & \rho_k \frac{1}{y_k^\top s_k} \\ & y_k = \nabla f_{k+1} - \nabla f_k \\ & s_k = x_{k+1} - x_k \end{aligned}$$

However, we need B_k^{-1} in $p_k = -B_k^{-1} \nabla f_k$, can we update $H_k = B_k^{-1}$ instead? Yes, just rewrite formula as

$$B_{k+1} = B_k + \begin{bmatrix} B_k s_k & y_k \end{bmatrix} \begin{bmatrix} 0 & -\rho_k \\ -\rho_k & \rho_k + s_k^\top B_k s_k \rho_k^2 \end{bmatrix} \begin{bmatrix} s_k^\top B_k \\ y_k^\top \end{bmatrix}.$$

Use the matrix inversion lemma (Sherman-Morrison-Woodbury formula)

$$(A + UCV)^{-1} = A^{-1} - U(C^{-1} + VA^{-1}U)^{-1} VA^{-1}$$

to obtain the inverse DFP formula:

$$H_{k+1} = H_k - \frac{H_k y_k y_k^\top H_k}{y_k^\top H_k y_k} + \frac{s_k s_k^\top}{y_k^\top s_k}.$$

5 BFGS update formula

As time went by, people agreed that BFGS update formula is better than DFP. Alternatively: Choose H_{k+1} closest to H_k:

$$H_{k+1} = \arg\min_H ||H - H_k|| \qquad \text{s.t.} \qquad H = H^\top,\ Hy_k = s_k.$$

Again we use the weighted Frobenius norm. Now the solution is the BFGS formula:

$$H_{k+1} = (I - \rho_k s_k y_k)^\top H_k (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top.$$

Very similar to the DFP, but considered the most effective Quasi-Newton formula. Note: H_{k+1} positive definite if $y_k^\top s_k > 0$.

BFGS algorithm:

```
Given starting point x_0, convergence tolerance ε > 0,
  inverse Hessian approximation H_0;
k = 0;
while ||∇f_k|| > ε;
  Compute search direction

    p_k = −H_k ∇f_k;
  Set x_{k+1} = x_k + α_k p_k where α_k is computed from a line search
    procedure to satisfy the Wolfe conditions
  Define s_k = x_{k+01} − x_k and y_k = ∇f_{k+1} − ∇f_k;
  Compue H_{k+1} by means of (that long formula)
  k++
end
```