Numerical Optimization
Assignment 5

Xingrong Zong, tnd179
Hold 03, Group 8

March 14th, 2022

Introduction

This assignment implements Broyden-Fletcher-Goldfarb-Shannon (BFGS) algorithm and analyses the invariance properties of algorithms. Also, using algorithms to show that Newton's Method is invariant to all invertible linear transformations of the objective function.

Theoretical exercises

Exercise 1

We assume that Newton's Method with backtracking line search is running on f with starting point x_0, producing a sequence of iterates $x_1, \cdots, x_T$ with step candidates $p_1, \cdots, p_T$.
Further, assume that we run the algorithm $g(x) = f(Ax)$, where A is square and invertible and choose as starting-point $\tilde{x}_0 = A^{-1}x_0$ leading to iterates $\tilde{x}_1, \cdots, \tilde{x}_T$ with step candidates $\tilde{p}_1, \cdots, \tilde{p}_T$.
First, we need to get the Hessian of this algorithm:

$$g(x) = f(Ax)$$
$$\nabla g(x) = A^\intercal[\nabla f(Ax)]$$
$$\nabla^2 g(x) = A^\intercal[\nabla^2 f(Ax)]A$$

Then, we assume that $\tilde{x}_k = A^{-1}x_k$. The search direction often has the form:

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

where B_k is a symmetric and nonsingular matrix. In the steepest descent method, B_k is simply the identity matrix I, while in Newton's method, B_k is the exact Hessian $\nabla^2 f(x_k)$.

$$\begin{aligned}
p_k &= -\nabla^2 f_k^{-1} \nabla f_k \\
\tilde{p}_k &= -\nabla^2 g(\tilde{x}_k)^{-1} \nabla g(\tilde{x}_k) \\
&= -(A^\intercal [\nabla^2 f(A\tilde{x}_k)]A)^{-1} A^\intercal [\nabla f(A\tilde{x}_k)] \\
&= -A^{-1}[\nabla^2 f(A\tilde{x}_k)]^{-1}(A^\intercal)^{-1} A^\intercal [\nabla f(A\tilde{x}_k)] \\
&= -A^{-1}[\nabla^2 f(A\tilde{x}_k)]^{-1}[\nabla f(A\tilde{x}_k)] \\
&= A^{-1} p_k
\end{aligned}$$

Since each iteration of a line search method computes a search direction p_k and then decides how far to move along that direction. The iteration is given by

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

where the step length α_k is a positive scalar.

$$\begin{aligned}
g(\tilde{x}_k + \alpha \tilde{p}_k) &= f(A\tilde{x}_k + A\alpha\tilde{p}_k) \\
&= f(AA^{-1}x_k + A\alpha A^{-1} p_k) \\
&= f(x_k + \alpha p_k)
\end{aligned}$$

Also, in backtracking line search,

$\bar{\alpha} > 0, \rho \in (0,1), c \in (0,1)$; Set $\alpha \leftarrow \bar{\alpha}$;

repeat until $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^\intercal p_k$(Wolfe conditions)

$\alpha \leftarrow \rho\alpha$;

end (repeat)

Terminate with $\alpha_k = \alpha$

Because Wolfe conditions are scale-invariant, where the step length α_k is a positive scalar and c is constant. We conclude that the step lengths α_k remain the same.

By referring back to week 3 assignment: we assumed that Steepest Descent with backtracking line search is running on function f with starting point x_0, producing a sequence of iterates $x_1, \cdots, x_T$. Then, we assumed to run the algorithm on g, with A is an orthogonal matrix and starting point $\tilde{x}_0 = A^\intercal x_0$. We concluded that the sequence of iterates satisfies $\tilde{x}_k = A^\intercal x_k$ and $g(\tilde{x}_k) = f(x_k)$.

For this algorithm we can conclude that assumption $\tilde{x}_k = A^{-1}x_k$ holds by induction and thus

$$\begin{aligned} g(\tilde{x}_k) &= f(A\tilde{x}_k) \\ &= f(AA^{-1}x_k) \\ &= f(x_k) \end{aligned}$$

Now we get the result of Newton's method is invariant to all invertible linear transformations of the objective function.

To make BFGS invariant to invertible linear transformations, it is necessary for the objectives $\min_B ||B - B_k||$ and $\min_H ||H - H_k||$ to be invariant under the same transformations. A norm that allows easy solution of the minimization problem and gives rise to a scale invariant optimization method is the weighted Frobenius norm

$$||A||_w \equiv ||W^{\frac{1}{2}} A W^{\frac{1}{2}}||_F$$

The choice of the weighting matrices W used to define the norms in $\min_B ||B - B_k||$ and $\min_H ||H - H_k||$ ensures that this condition holds.

Programming exercises

Implementation

We implement BFGS, Algorithm 6.1:

> Given starting point x_0, convergence tolerance $\in > 0$,
>
> inverse Hessian approximation H_0;
>
> $k \leftarrow 0$;
>
> **while** $||\nabla f_k|| > \in$;
>
> Compute search direction
>
> $$p_k = -H_k \nabla f_k;$$
>
> Set $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed from a line search procedure to satisfy the Wolfe condition;
>
> 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
>
> $$H_{k+1} = (I - \rho_k s_k y_k^\intercal) H_k (I - \rho_k y_k s_k^\intercal) + \rho_k s_k s_k^\intercal;$$
>
> $k \leftarrow k + 1$;
>
> **end (while)**

Also, we implement Wolfe condition, Algorithm 3.6, in backtracking line search to get α:

> $\bar{\alpha} > 0, \rho \in (0, 1), c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$;
>
> **repeat** until $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^\intercal p_k$(Wolfe conditions)
>
> $\alpha \leftarrow \rho\alpha$;
>
> **end (repeat)**
>
> Terminate with $\alpha_k = \alpha$

Conclusion

This assignment gave me a more complete understanding of the connections between Newton's Method and Broyden-Fletcher-Goldfarb-Shannon (BFGS) algorithms, and the properties they both have. Also, implemented BFGS procedure on previous algorithms.

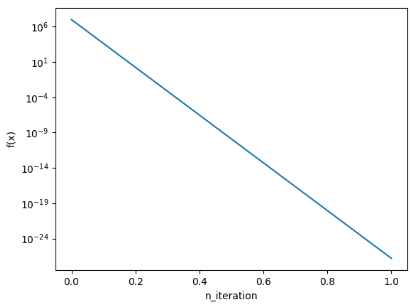

Figure 1: Ellipsoid Function

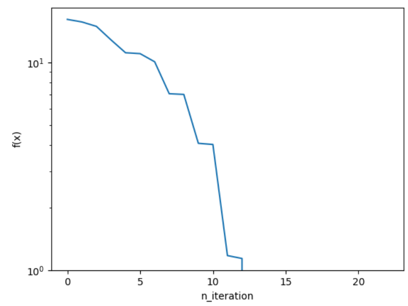

Figure 2: Log-Ellipsoid Function

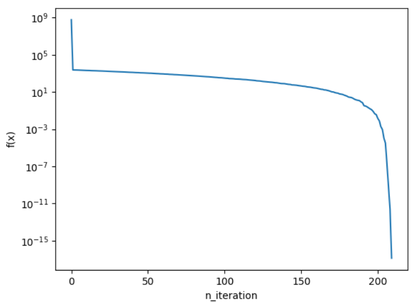

Figure 3: Rosenbrock Banana Function

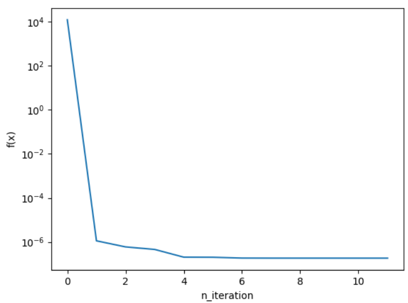

Figure 4: Attractive-Sector Function

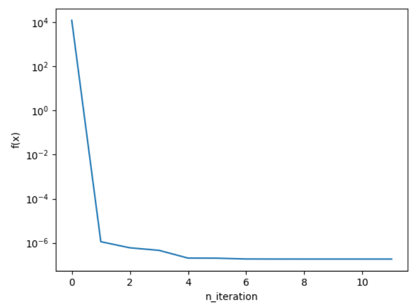

Figure 5: Sum of Different Powers Function

	Ellipsoid	Log-Ellipsoid	Rosenbrock Banana	Attractive-Sector	Sum of Different Powers
x	[0.0 7.105e-15 0.0]	[0.0 6.199e-13 6.373e-15]	[1.0 1.0]	[1.908e-4 1.918e-4 1.907e-4]	[0.0 0.011]
f(x)	1.597e-27	-9.210	1.310e-17	1.831e-07	1.614
n-iteration	1	22	209	11	32