AMATH 515: HW4

NGHI NGUYEN

Applied Math Department, University of Washington, Seattle, WA
nghiiuu@uw.edu

1. Introduction

Here we investigate the effects of four different optimization schemes on the Rosenbrock function, the Logistic Regression function, and the Sum Squares Function. Specifically, the Steepest Descent, Newton,David-Fletchen-Powell (DFP), and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method. Each method will be implemented with the Bisection Line Search with Wolfe conditions. With these four implementation, the goal of this paper is compare find the error convergence to the tolerance with respect to the number of iterations and seconds it takes.

2. Methods

The Steepest Descent, Newton, David-Fletchen-Powell (DFP), and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm are used to find the minimum value of the Rosenbrock function

$$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 + 100\sum_{k=1}^{n-1}(x_i^2 - x_{i+1})^2 \tag{1}$$

, the Regression Logistic function

$$f(x) = \sum_{i=1}^{m}\{\log(1 + \exp(\langle a_i, x\rangle)) - b_i\langle a_i, x\rangle\} + \frac{\lambda}{2}\|x\|^2. \tag{2}$$

, and the Sum Squares function

$$f(x) = \sum_{i=1}^{n} i * x^2 \tag{3}$$

Each algorithms are implemented with the Bisection Line Search with the Wolfe Conditions, condition 1,

$$W_1(t) := \frac{f(x^k + td^k) - f(x^k)}{t(d^k)^T\nabla f(x^k)} > c1$$

and condition 2

$$W_2(t) := \frac{(d^k)^T\nabla f(x^k + td^k)}{(d^k)^T\nabla f(x^k)} < c2$$

. Condition 1 prevents the step size from being too large, whereas condition 2 prevents the step size from being too small. The Bisection Line Search function is called in each algorithm with different x and d values that are unique to the algorithms, returning an optimal step-size for each method. The parameters $c1 = .3, /c2 = .4$ for both Rosenbrock and Sum Squares function, where $c1 = .1, /c2 = .4$ for the Logistic Regression function

Date: February 18, 2025.

2.1. **Algorithms.** For all of the algorithm, the next x value after each iteration is $x^(k+1) = x^k + \alpha^k * d * k$ where α is the step size from the Bisection Line Search and d is the direction search.

The **Steepest Descent** uses the opposite direction of the gradient of $f(x)$, $\nabla f(x^k)$ as the direction search $d^k = -\nabla f(x^k)$
The **Newton** method uses both the Hessian matrix, $\nabla^2 f(x)$, and the gradient, $\nabla f(x)$, of $f(x)$ for the search direction.$d^k = -\nabla^2 f(x^k)\nabla f(x^k)$
Here, the Hessian matrix is computed directly for both 1 and 2.

The **DFP** and **BFGS** methods are both Quasi-Newton Methods, a modification of the Newton Method, that approximates the Hessian Matrix, $B(x^k)$ using the gradient of the function. The Hessian Matrix is computed recursively. The Hessian Matrix approximation for the k+1 step for DFP,

$$B(x^{k+1}) = B(x^k) - \frac{(B(x^k)y^k(B(x^k)y^k)^T}{(y^k)^T B(x^k)y^k} + \frac{s^k(s^k)^T}{(y^k)^T s^k}$$

for BFGS,

$$B(x^{k+1}) = (I - \frac{s^k(y^k)^T}{(y^k)^T s^k})B(x^k)(I - \frac{y^k(s^k)^T}{(y^k)^T s^k}) + \frac{s^k(s^k)^T}{(y^k)^T s^k}$$

Comparing both, the Newton Method has a O(n^3) versus DFP and BFGS have an O(n^2) cost for computing the Hessian Matrix. The affect of this cost difference on convergence rate is mentioned more in the 3.

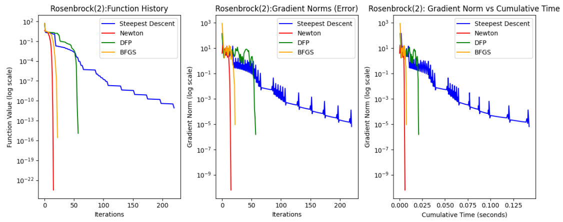
3. Results

For the Rosenbrock Function, the error for the the Newton Method converges the fastest with the least amount of iteration, whereas the error of the Steepest Descent Method converges the slowest with the most amount of iteration for all values of n (2, 5, 10, and 50). The Steepest Descent method requires a higher amount of iteration to minimize the error because the Steepest Decent method is a lower-order approximation given it is a linear approximation, whereas the Newton Method is a higher-order approximation as it is a quadratic approximation.
For the Quasi-Newton method, both of the errors for DFP and BFGS converges to the tolerance in between the number of iterations required from the Newton method and Steepest Descent method.

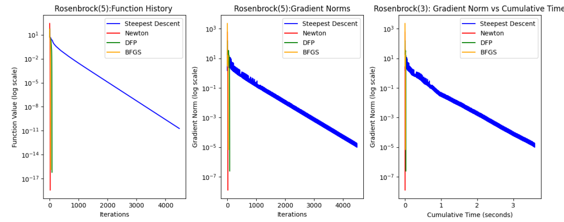
The third graph for each sub-figure shows how fast each method convergence for the respective function and algorithm. For all of the Rosenbrock Functions (n=2, 5, 10, 50), the convergence order of each algorithm is the same with respect to iteration and time: the Newton method is fastest in second, whereas the Steepest Descent method is the slowest time. In theory, the Newton method should be slower than the Quasi-Newton methods as the Newton method uses the inverse function to compute the Hessian matrix, a more expensive approach. Here, Newton is still faster than both Quasi-Newton method as the dimension n only ranges 2 to 50 dimensions. However, both Quasi-Newton method does approach the time length of the Newton method as n increase, an indicator Newton is more expensive than both Quasi-Newton Method.

You may also need to include multiple figures:

For the Regressional Logistic Function, the Newton method requires the least amount of iterations for error convergence. However, the DFP method requires the most iterations for $\lambda = .001$ and the Steepest Descent method requires the most iterations for $\lambda = .01, .1$. This difference in

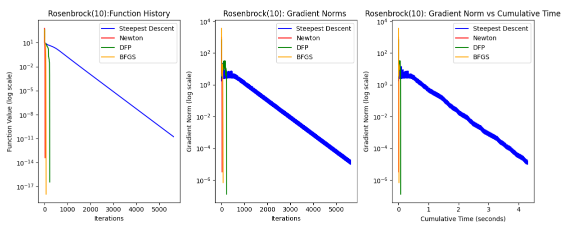


(A) Rosenbrock(2)

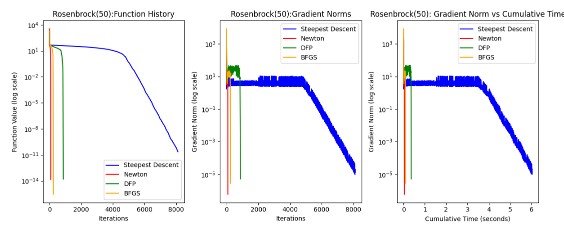


(B) Rosenbrock(5)

FIGURE 1. A and B plots the function values and gradient norm (error) with respect to iterations and the gradient norm (error) with respect cumulative time .



(A) Rosenbrock(10)

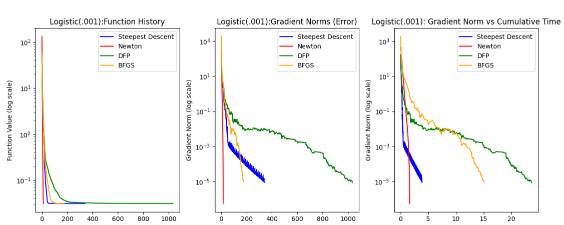


(B) Rosenbrock(50)

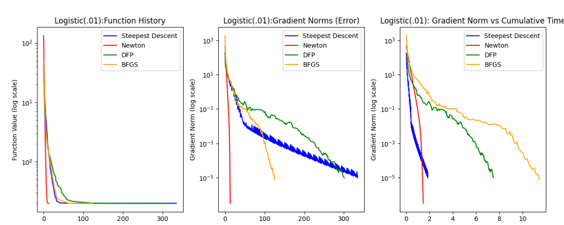
FIGURE 2. A and B plots the function values and gradient norm (error) with respect to iterations and the gradient norm (error) with respect cumulative time .

the convergence order for the Regresstional Logistic Function versus the Rosenbruck Function is potentially stemming from the dimension of the matrix A in the Regresstional Logistic function, a dimension of 784.

In addition, the Steepest Descent method is almost as fast as the Newton method to converge to the tolerance. Contrary to the Rosenbrock function, as it was the slowest to converge.

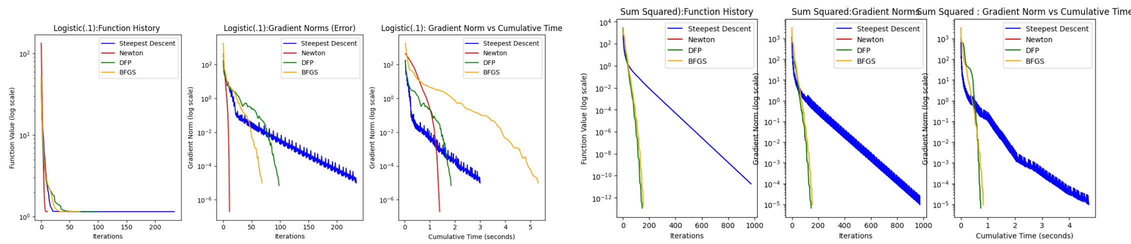


(A) Logistic Regression(.001)



(B) Logistic Regression(.01)

FIGURE 3. A and B plots the function values and gradient norm (error) with respect to iterations and the gradient norm (error) with respect cumulative time .



(A) Logistic Regression(.1)

(B) The Sum Squares

FIGURE 4. A and B plots the function values and gradient norm (error) with respect to iterations and the gradient norm (error) with respect cumulative time .

The third test problem is the Sum Squares Function. The error for Newton method converges in one step since the the order of convergence for the Newton is quadratic and the objective function is a quadratic. The Steepest Descent takes the longest to converge with respect to iterations and time.

4. Summary and Conclusions

In conclusion, the errors for the Newton method generally converges the fastest with respect to iterations and time, whereas the Steepest Descent generally converges the slowest with iterations and the Quasi-Newton Method converges the slowest for time.

Acknowledgments

I recieed help from Emily Zhang, Johan Azambou, Crimson Stambaugh, Cade Ballew, and Alex Hsu.