Report on Project 2 - Unconstrained Optimization

By
Muhammad Raees (mr2714@rit.edu)
Sujan Dutta (sd2516@rit.edu)

1. Overview

The objective of this project is to implement unconstrained optimization for given three objective functions. We implemented the three required methods for optimization (Gradient Descent, Newton, and Quasi-Newton) and two additional methods (Marquardt, and Polak-Ribiere Conjugate Gradient). In most of the test runs, gradient descent has the slowest convergence while the Polak-Ribiere conjugate gradient is the second worst. The Newton method is faster to converge but it may have higher costs due to the computations of the hessian much like the Marquardt method which is a mixture of the gradient descent and newton. Polak-Ribiere is a variation of the nonlinear Fletcher-Reeves Conjugate Gradient method. The methods are tested with various values of alpha in the range (0, 1). Increasing the alpha's value affects the Newton, Quasi-Newton, and Marquardt methods positively. The Quasi-Newton gets negligibly slower in function 2 with an increase in alpha. No trend is observed in the gradient descent with an increase in the value. The results are reported on the following parameters.

α = 1; tolerance = 0.000001; maximum iterations=1000.
c1 = 0.2; c2 = 0.5; (Line search parameters)

2. Optimization Methods

2.1 Function 1

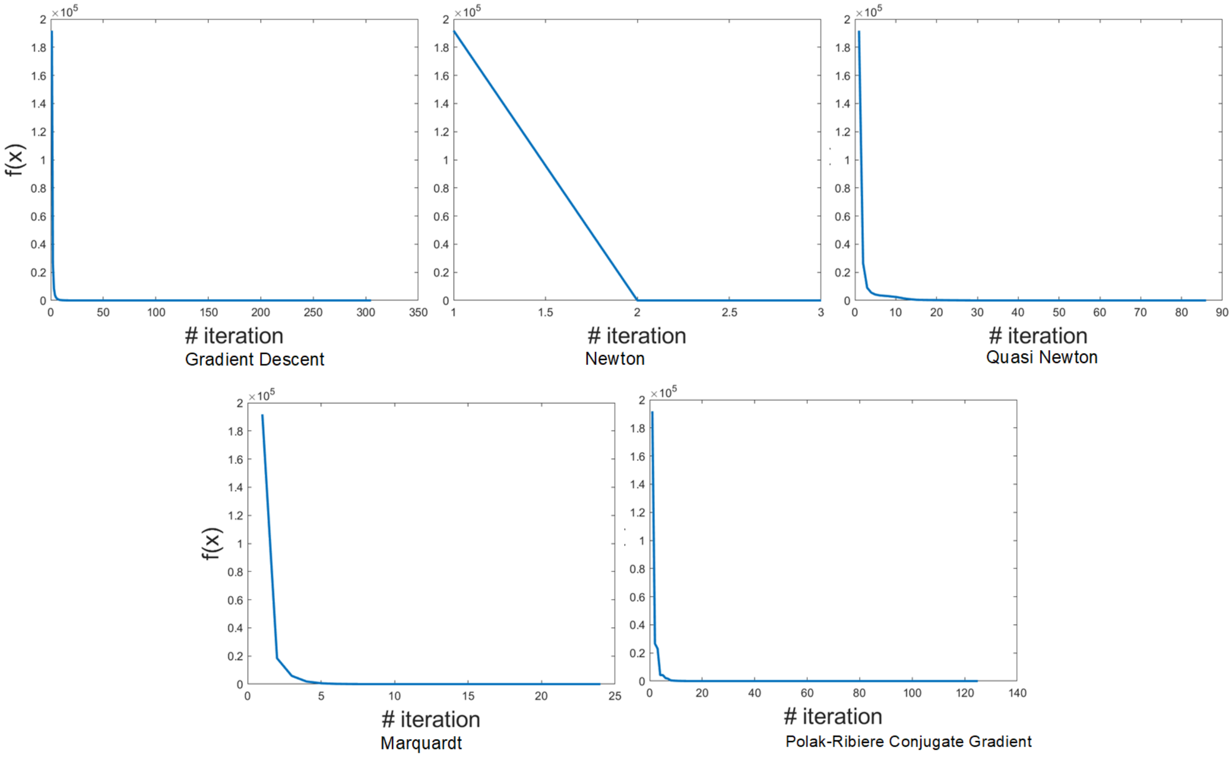
$f(\mathbf{x}) = \sum_{i=1}^{n} (i \cdot x_i^2)$, where $n = 100$. Global minimum $f(\mathbf{x})_{\min} = 0$.

Table 1 shows a comparison of the optimization results for function 1. The function values are compared and reported on two initial values of inputs (100 sequential numbers in the range from -49 to 50, and random 100 numbers between 1 to 10). In both cases, the Newton method converges to the analytical solution in one iteration. This is because the function is quadratic in nature. Marquardt is second best with a low number of iterations. Gradient descent is simple but the slowest to converge due to its dependence on the scale of the problem and input. The conjugate gradient lies between Newton/Quasi-Newton and Gradient descent, and it does not require hessian. Figure 1 presents the comparison for the convergence of the function value at iterations concerning the global minimum of function 1 (i.e., 0) for all the five participating optimization methods.

Report on Project 2 - Unconstrained Optimization

Table 1. Optimization results for function 1 with different values of input.

Sr.	*Optimization Method*	*X= [-49:1:50]*		*X= randi ([1 10],1,100)*	
		Min(f(x))	*Iterations*	*Min(f(x))*	*Iterations*
1	Gradient Descent	2.5979e-05	412	3.0451e-05	236
2	Newton	0	1	0	1
3	Quasi-Newton	1.7504e-05	96	2.8689e-06	84
4	Marquardt	3.9358e-07	7	9.6820e-07	5
5	Polak-Ribiere Conjugate Gradient	2.8671e-15	149	7.5017e-15	106



Gradient Descent

Newton

Quasi Newton

Marquardt

Polak-Ribiere Conjugate Gradient

Figure 1. The comparison of function value at each iteration with each method.

2.2 Function 2

$f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} - \sum_{i=1}^{m} \log(b_i - \mathbf{a}_i^T\mathbf{x})$ where $m = 500$ and $n = 100$.

Table 2 presents the optimization results for function 2. The function values are reported on the initial zero inputs (a vector of 100 zeros i.e., X= zeros (100,1)). The function is only defined when the following condition is met: $b_i - a_i^T x > 0$ for i. Hence before updating x at each iteration, we check if this condition holds. The rest of the mechanism is the same where the calculation does not need the constraint. The results indicate that gradient descent and conjugate gradient need more steps to reach the same minimum obtained by others in fewer steps. Newton and Marquardt using hessian can reach the same conclusion faster.

Report on Project 2 - Unconstrained Optimization

Table 2. Optimization results for function 2.

Sr.	***Optimization Method***	***X= zeros (100,1)***	
		Min(f(x))	***Iterations***
1	Gradient Descent	-2.4431e+03	1000
2	Newton	-2.4432e+03	9
3	Quasi-Newton	-2.4432e+03	174
4	Marquardt	-2.4432e+03	12
5	Polak-Ribiere Conjugate Gradient	-2.4431e+03	1000

2.3 Function 3

$$f(\mathrm{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

The global minima (0) of the function occur at point (1, 1) and all methods behave similarly on the input X = [1, 1]. Table 3 shows a comparison of the optimization results for function 3. The function is more sensitive to the input for gradient descent and takes a longer time to converge. The newton method is again the fastest and quasi-newton performs better than the Marquardt method. The function values are compared and reported on two initial values of inputs (two points chosen randomly).

Table 3. Optimization results for function 3 with different values of input.

Sr.	***Optimization Method***	***X= [1.3;1.5]***		***X= [2;1]***	
		Min(f(x))	***Iterations***	***Min(f(x))***	***Iterations***
1	Gradient Descent	1.0119e-03	632	5.9693e-04	183
2	Newton	3.6556e-07	7	9.2788e-08	13
3	Quasi-Newton	2.5628e-07	8	2.6687e-07	17
4	Marquardt	4.7810e-07	13	1.3560e-06	19
5	Polak-Ribiere Conjugate Gradient	1.1739e-12	332	1.2111e-12	841

3. Conclusion

The objective functions are optimized using the five optimization methods. The methods with hessian computations are faster to converge. The learning rate does affect the convergence rate of complex algorithms like newton, quasi-newton, and the Marquardt method. However, the effect cannot be generalized to gradient descent and conjugate descent methods. Function 2 is troublesome as it requires the constraint $b_i - a_i^T x > 0$. Because of this constraint the discussed methods are not ideal for the optimization of this function. The performance of methods does not vary much which also hints at a limitation on optimization for function 2.