Limited-Memory Quasi-Newton Optimization Algorithm

INTRODUCTION

Given a model and dataset, parameter estimation reduces to solving an unconstrained optimization problem:

$$x^* = \arg\min_x f(x) \tag{1}$$

where f is a convex, twice differentiable objective function, and $x \in \mathbb{R}^n$.

The quadratic approximation using Taylor expansion is:

$$f(x + \Delta x) \approx f(x) + \Delta x^T \nabla f(x) + \frac{1}{2}\Delta x^T (\nabla^2 f(x))\Delta x$$

For Δx such that $\nabla f(x + \Delta x) = 0$, Newton's updates:

$$x_{n+1} = x_n - t \cdot (\nabla^2 f(x))^{-1} \nabla f(x) \tag{2}$$

Pros of Newton's method:

Rapid convergence (typically quadratic), robust and trustworthy.

Cons of Newton's method:

Heavy computations of Hessian, takes $\mathcal{O}(n^3)$ operations. Also, the Hessian may not be invertible or may be ill-conditioned, leading to numerical instabilities

Test problem: Rosenbrock function

The function is given by:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2$$

It exhibits a global minimum at $(x, y) = (a, a^2)$

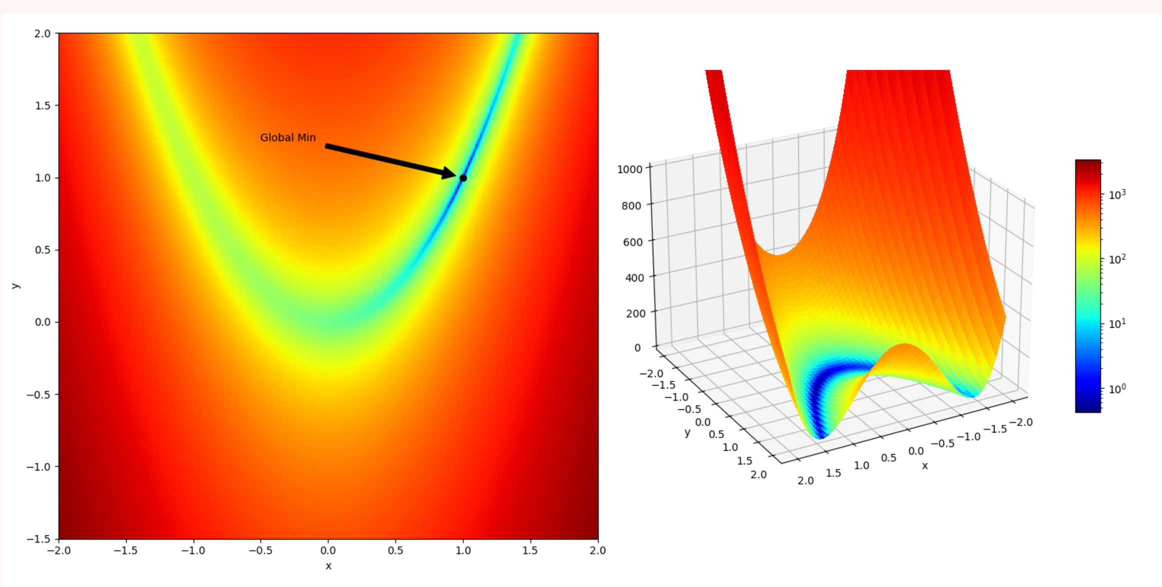

Figure 1. Contour plot of Rosenbrock function

APPROXIMATED HESSIAN..?

Let B_k be the approximation of the inverse Hessian matrix at iteration k. The **BFGS** updates this approximation using the following recurrence relation: [2]

$$B_{k+1} = (I - \rho_k s_k y_k^T) B_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T \tag{3}$$

Where:

s_k is the step vector, representing the change in the solution from iteration k to $k+1$

y_k is the change in the gradient vector from iteration k to $k+1$

$\rho_k = \frac{1}{y_k^T s_k}$

I is the identity matrix

The BFGS update is still quite cheap: $\mathcal{O}(n^2)$ operations.

LIMITED MEMORY-BFGS

L-BFGS implicitly stores a modified version of H_k using a limited number m of recent vector pairs (s_i, y_i). This allows efficient computation of $B_k \nabla f_k$ through vector operations, while replacing the oldest pair with the new (s_k, y_k) after each iteration.

This brings down the cost of each update to $\mathcal{O}(mn)$ operations. This is great because modest values of 'm' produce satisfactory results.

Psuedocode

$q \leftarrow \nabla f_k$;

for $i = k-1, k-2, \dots, k-m$ **do**
  $\alpha_i \leftarrow \rho_i s_i^T q$
  $q \leftarrow q - \alpha_i y_i$
end for
$r \leftarrow H_0^k q$
for $i = k-m, k-m+1, \dots, k-1$ **do**
  $\beta \leftarrow \rho_i y_i^T r$
  $r \leftarrow r + s_i(\alpha_i - \beta)$
end for
stop with result $H_k \nabla f_k \quad r$ =0

ANALYSIS

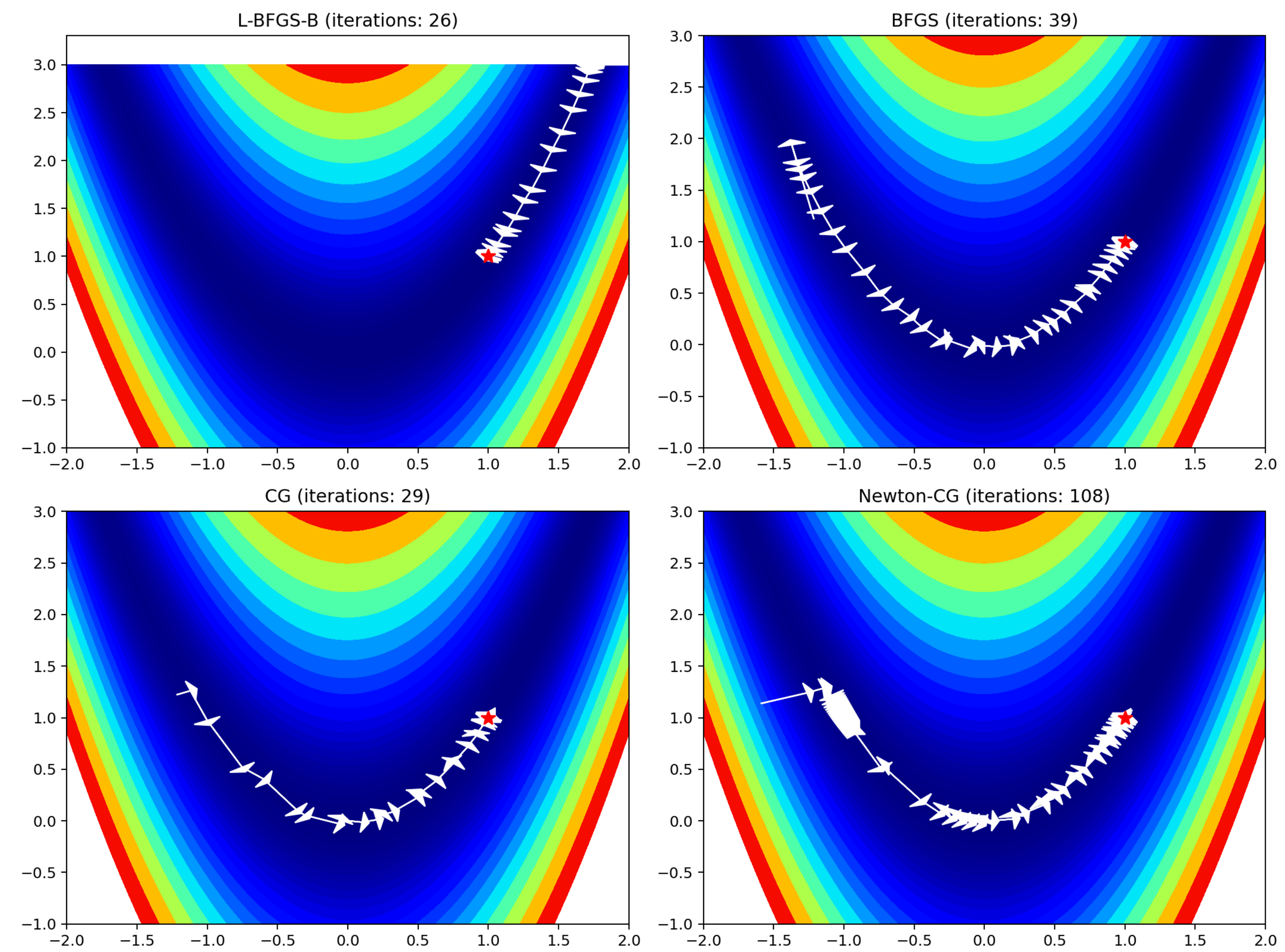

Figure 2. Path plots of various optimization algorithm for determining global minima of Rosenbrock function, with initial guess $(-2.2, 1.0)$

Algorithm	Average Runtime	Average iterations
L-BFGS	**0.002452**	**26**
BFGS	0.005413	39
CG	0.006460	29
Newton-CG	0.023046	108

Table 1. Results of the comparative study

REFERENCES

[1] Richard H. Byrd, Jorge Nocedal, and Ya-Xiang Yuan. Global convergence of a class of quasi-newton methods on convex problems. *SIAM Journal on Numerical Analysis*, 24(5):1171–1190, 1987.

[2] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

Abhishek Singh
CS460 MACHINE LEARNING POSTER SESSION
SCoS, NISER