Quasi-Newton Methods

Ankesh Kumar

1 Newton's Method

Newton's method is an iterative optimization technique that uses the second-order derivative information (Hessian matrix) of the objective function $f(x)$ to find its minimum. At each iteration, it updates the current estimate x_k of the minimum using the formula:

$$x_{k+1} = x_k - H^{-1}(x_k)\nabla f(x_k)$$

where:

- $\nabla f(x_k)$ is the gradient of f at x_k.
- $H(x_k)$ is the Hessian matrix of f at x_k.

Limitations of Newton's Method

- **Computational Cost**: Computing and inverting the Hessian matrix $H(x_k)$ can be computationally expensive, especially for large-scale problems.
- **Storage Requirements**: The Hessian matrix $H(x_k)$ is typically large and dense, requiring significant memory resources.

2 Quasi-Newton Methods

Quasi-Newton methods address these limitations by approximating the Hessian matrix $H(x_k)$ using information gathered during the optimization process, without explicitly computing it. Instead of using the exact Hessian, they use an approximation B_k of the inverse Hessian $H^{-1}(x_k)$.

Update Rule

At each iteration k, the update rule for Quasi-Newton methods typically takes the form:

$$x_{k+1} = x_k - \alpha_k B_k \nabla f(x_k)$$

where:

- B_k is an approximation to $H^{-1}(x_k)$.
- α_k is a step size or line search parameter.

Difference from Newton's Method

- **Approximation of Hessian**: Quasi-Newton methods do not require the explicit computation or inversion of the Hessian matrix $H(x_k)$. Instead, they iteratively update an approximation B_k of $H^{-1}(x_k)$.
- **Computational Efficiency**: By avoiding direct Hessian computations, Quasi-Newton methods can be more computationally efficient and are particularly suited for problems where the Hessian is costly to compute or store.

3 Inverse Hessian Context

The inverse Hessian $H^{-1}(x_k)$ plays a crucial role in both Newton's method and Quasi-Newton methods:

- **Newton's Method**: It directly computes and uses $H^{-1}(x_k)$ to determine the search direction.
- **Quasi-Newton Methods**: They use an approximation B_k of $H^{-1}(x_k)$, updated iteratively based on the gradient differences between successive points.

4 Rank One Correction (SRS) Algorithm

The Rank One Correction, or Single-Rank Symmetric (SRS) update, is a quasi-Newton method that iteratively updates the approximation of the Hessian matrix B_k using gradient differences. This method aims to enhance optimization efficiency by approximating the Hessian without directly computing it.

4.1 Update Rule

At each iteration k, the SRS algorithm updates the approximation B_k of the Hessian using the formula:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k}$$

where:

- $s_k = x_{k+1} - x_k$ is the difference in variable updates.
- $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ is the difference in gradient evaluations.

4.2 Explanation

- The SRS update maintains symmetry of the approximated Hessian B_k.
- It modifies B_k by adding a rank-one matrix $\frac{y_k y_k^T}{y_k^T s_k}$, where y_k is the change in gradient and s_k is the change in variables.
- The denominator $y_k^T s_k$ should be non-zero for the update to be well-defined and avoid division by zero.
- Unlike some other quasi-Newton methods, SRS does not guarantee positive definiteness of B_k.

4.3 Advantages

- **Computational Efficiency**: SRS avoids direct computation of the Hessian matrix, making it computationally efficient.
- **Accurate Approximation**: It can provide a good approximation of the true Hessian and improve convergence rates in many optimization problems.
- **Simplicity**: The update rule involves a straightforward rank-one update, which is easy to implement and understand.

4.4 Challenges

- **Numerical Stability**: The SRS update may face numerical instability if the denominator $y_k^T s_k$ is close to zero.
- **Non-Positive Definiteness**: Similar to other quasi-Newton methods, SRS does not ensure positive definiteness of B_k, which can affect optimization performance in certain scenarios.

4.5 Application

The SRS update is commonly used in optimization algorithms, particularly in trust-region methods and other applications where an efficient approximation of the Hessian is crucial. It strikes a balance between computational efficiency and accuracy, making it suitable for large-scale optimization problems.

5 DFP Algorithm

5.1 Objective

The DFP algorithm aims to minimize a continuously differentiable objective function $f : \mathbb{R}^n \to \mathbb{R}$ using an iterative approach that approximates the inverse Hessian matrix $H^{-1}(x_k)$.

5.2 Iterative Update

- **Initialization:** Start with an initial positive definite approximation B_0 of $H^{-1}(x_0)$, often chosen as the identity matrix I.
- **Update Rule:** At each iteration k, update B_k using the formula:

$$B_{k+1} = B_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{B_k y_k y_k^T B_k}{y_k^T B_k y_k}$$

  where:

  - $s_k = x_{k+1} - x_k$ is the vector of changes in the parameter vector x.
  - $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ is the difference in gradients at iterations $k+1$ and k.

6 BFGS Algorithm

6.1 Objective

The BFGS algorithm aims to minimize a continuously differentiable objective function $f : \mathbb{R}^n \to \mathbb{R}$ using an iterative approach that approximates the inverse Hessian matrix $H^{-1}(x_k)$.

6.2 Iterative Update

- **Initialization:** Start with an initial positive definite approximation B_0 of $H^{-1}(x_0)$, often chosen as the identity matrix I.
- **Update Rule:** At each iteration k, update B_k using the formula:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}$$

  where:

  - $s_k = x_{k+1} - x_k$ is the vector of changes in the parameter vector x.
  - $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ is the difference in gradients at iterations $k+1$ and k.

7 Comparison

- Both DFP and BFGS are Quasi-Newton methods for unconstrained optimization.
- **Update Formulas:** DFP updates B_k using s_k and y_k, while BFGS updates B_k using y_k and s_k.

- **Positive Definiteness:** BFGS guarantees B_k remains positive definite, whereas DFP does not inherently maintain this property.
- **Convergence:** BFGS generally exhibits faster convergence rates due to its positive definiteness maintenance.
- **Computational Efficiency:** Both methods are efficient alternatives to computing the full Hessian matrix.

8 Discussion of Quasi-Newton Method Solutions

In this section, we discuss the solutions obtained from implementing various quasi-Newton methods on the Rosenbrock function. The methods used were BFGS, SR1, DFP, and an approximation of the inverse Hessian. The initial condition was $\mathbf{x_0} = [-2, 2]^T$, and the maximum number of iterations was set to 10,000.

8.1 Results

The table below summarizes the optimal solutions and function values obtained from each method:

Method	**Optimal Solution**	**Optimal Function Value**
BFGS	$[1.0000, 1.0000]^T$	3.2463×10^{-13}
SR1	$[-1.9400, 2.0152]^T$	314.2801
DFP	$[1.0000, 1.0000]^T$	2.9002×10^{-20}
Approx. Inv. Hessian	$[-1.9400, 2.0152]^T$	314.2802

Table 1: Optimal solutions and function values for different quasi-Newton methods on the Rosenbrock function.

8.2 Analysis

The BFGS and DFP methods both successfully converged to the global minimum of the Rosenbrock function, located at $\mathbf{x}^* = [1, 1]^T$, with function values approaching zero. This indicates that both methods are highly effective for this optimization problem, showing fast convergence and high accuracy.

On the other hand, the SR1 and Approximate Inverse Hessian methods did not converge to the global minimum. Instead, they found suboptimal solutions at $\mathbf{x} = [-1.9400, 2.0152]^T$ with significantly higher function values around 314. This discrepancy suggests that these methods might not be as robust or effective for the Rosenbrock function, potentially due to numerical instability or the lack of guaranteed positive definiteness in the case of SR1.

8.3 Comparison

The comparison between these methods reveals that BFGS and DFP are more reliable for optimizing the Rosenbrock function, both achieving solutions very close to the true global minimum. SR1 and the Approximate Inverse Hessian method, while sometimes effective, may struggle with certain functions due to their respective limitations. In practice, choosing the appropriate quasi-Newton method can significantly impact the efficiency and accuracy of the optimization process.

9 Summary

- **Newton's Method**: Computes and uses the exact Hessian $H(x_k)$ for optimization.
- **Quasi-Newton Methods**: Approximates the inverse Hessian $H^{-1}(x_k)$ iteratively, avoiding direct computation and inversion of $H(x_k)$.