NEWTON TYPE METHODS

Aysen YILDIZ
Esra Nur CAKICI
H.Yavuz ERZURUMLU

May 2, 2018

Abstract

In this paper we studied on newton type methods which is the most famous approach for solving nonlinear equations and minimizing functions. This article gives a short introduction to newton type methods with a brief a discussion of some of the main issues in applying this algorithm for the solution of the practical problems. We compare three types of methods which use this approach.

1 Introduction

Newton's method is using for solving nonlinear equations and approximation for local convergence. Its easy to implement and barely fast. It requires only first and second derivatives. Lots of scientific and engineering applications use this approach widely. It must be modified for running more efficiently.

One of the advantages of Newton's method is very simple that means it is easy to implement. It also fast rate of convergence.

Despite these advantages , it has some drawbacks. Due to be a local method, you may not reach minimum or maximum value of function. Also you need first order information about the functions first and second derivatives. You need to solve several linear system.

There are several areas which adopt Newton's method approach:

- **Estimating root of f(x) function**: Newton Raphson Method
- **One Dimensional Unconstrained Optimization**: Newton's Method
- **Multi Dimensional Unconstrained Optimization**: Gradient Flow Method (With Hessian)

2 Analysis of Methods

2.1 Estimating Root of 1-D Function - Newton-Raphson Method

Perhaps the most widely used of all root-locating formulas is the Newton-Raphson equation. If the initial guess at the root is x_i, a tangent can be extended from the point $[x_i, f(x_i)]$. The point where this tangent crosses the x axis usually represents an improved estimate of the root.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \tag{1}$$

Algorithm:

- **Input:** initial x, $func(x), derivFunc(x)$
- **Output:** Root of Func()
  1. Compute values of $func(x)$ and $derivFunc(x)$ for given initial x
  2. Compute h: $h = \frac{func(x)}{derivFunc(x)}$
  3. While h is greater than allowed error ε
     (a) $h = \frac{func(x)}{derivFunc(x)}$
     (b) $x = x - h$

Example:

Let us illustrate Newton's method with a concrete numerical example. The golden ratio ($\varphi \approx 1.618$) is the largest root of the polynomial $f(x) = x^2 - x - 1$; to calculate this root, we can use the Newton iteration

$x_{k+1} = x_k - \frac{f(x)}{f'(x)} = x_k - \frac{x_k^2 - x_k - 1}{2x_k - 1} = \frac{x_k^2 + 1}{2x_k - 1}$

with the initial estimate $x_0 = 1$. Using double-precision floating-point arithmetic, which amounts to a precision of roughly 16 decimal digits, Newton's method produces the following sequence of approximations:

.

$x_0 = 1.0$
$x_1 = 2.0$
$x_2 = 1.6666666666666667$
$x_3 = 1.6190476190476191$
$x_4 = 1.6180344478216817$
$x_5 = 1.618033988749989$
$x_6 = 1.6180339887498947$
$x_7 = 1.6180339887498949$
$x_8 = 1.6180339887498949$

Since the last iteration does not change the value, we can be reasonably sure to have obtained a value for the golden ratio that is correct to the precision used.

Complexity:

Using Newton's method as described above, the time complexity of calculating a root of a function $f(x)$ with n-digit precision, provided that a good initial approximation is known, is $O((logn)F(n))$ where $F(n)$ is the cost of calculating $f(x)/f'(x)$ with n-digit precision.

2.2 One Dimensional Unconstrained Optimization

Recall that the Newton-Raphson method is an open method that finds the root x of a function such that $f(x) = 0$

$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$

as a technique to find the minimum or maximum of $f(x)$. It should be noted that this equation can also be derived by writing a second-order Taylor series for $f(x)$ and setting the derivative of the series equal to zero. Newtons method is an open method similar to Newton-Raphson because it does not require initial guesses that bracket the optimum. In addition, it also shares the disadvantage that it may be divergent. Finally, it is usually a good idea to check that the second derivative has the correct sign to confirm that the technique is converging on the result you desire.

Example :

Use Newtons method to find the maximum of

$f(x) = 2\sin(x) - \frac{x^2}{10}$

with an initial guess of $x_0 = 2.5$

Solution: The first and second derivatives of the function can be evaluated as

$f'(x) = 2\cos(x) - \frac{x}{5}$

$f''(x) = -2\sin(x) - \frac{1}{5}$

which can be subsituted into Eq. to give

$x_{i+1} = x_i - \frac{2\cos(x_i) - x_i/5}{-2\sin(x_i) - 1/5}$

Subsituting the initial guess yields

$x_i = 2.5 - \frac{2\cos 2.5 - 2.5/5}{-2\sin 2.5 - 1/5} = 0.99508$

which has a function value of 1.57859. The second iteration gives

$x_i = 0.0995 - \frac{2\cos 0.0995 - 0.0995/5}{-2\sin 0.0995 - 1/5} = 1.46901$

which has a function value of 1.77385.

The process can be repeated, with the results tabulated at Table Results.

i	x	$f(x)$	$f'(x)$	$f''(x)$
0	2.5	0.57194	-2.10229	-1.39694
1	0.99508	1.57859	0.88985	-1.87761
2	1.46901	1.77385	-0.09058	-2.18965
3	1.42764	1.77573	-0.00020	-2.17954
4	1.42755	1.77573	0.00000	-2.17952

Table 1: Results

Thus, within four iterations, the result converges rapidly on the true value.

Although Newtons method works well in some cases, it is impractical for cases where the derivatives cannot be conveniently evaluated. For these cases, other approaches that do not involve derivative evaluation are available. For example, a secant-like version of Newtons method can be developed by using finite-difference approximations for the derivative evaluations.

Complexity :

Complexity is related with the initial value. It is $O(lg(n))$ which n is initial value.

2.3 Multidimensional Unconstrained Optimization

2.3.1 Newton Gradient Methods

As the name implies, gradient methods explicitly use derivative information to generate efficient algorithms to locate optima.

$|H| > 0$ and $\partial^2 f/\partial x^2 > 0$, then $f(x, y)$ has a local minimum.
$|H| > 0$ and $\partial^2 f/\partial x^2 < 0$, then $f(x, y)$ has a local maximum.
$|H| < 0$, then $f(x, y)$ has a saddle point.

2.3.2 The Hessian

For one-dimensional problems, both the first and second derivatives provide valuable information for searching out optima. The first derivative a provide steepest trajectory of the function and b tells us that we have reached an optimum.

Once at an optimum, the second derivative tells us whether we are a maximum $[negative f\text{''}(x)]$ or a minimum $[positive f\text{''}(x)]$. Now, we will examine how the second derivative is used in such contexts.You might expect that if the partial second derivatives with respect to both x and y are both negative, then you have reached a maximum.

The quantity $|H|$ is equal to the determinant of a matrix made up of the second derivatives.

Newton's method for a single variable can be extended to multivariate cases. Write a second-order Taylor series for $f(x)$ near $x = x_i$,

$$f(x) = f(x_i) + \nabla f^T(x_i)(x - x_i) + \frac{1}{2}(x - x_i)^T H_i(x - x_i)$$

where H_i is the Hessian matrix. At the minimum,

$\frac{\partial f(x)}{\partial x_j} = 0$ for $j = 1, 2, ..., n$

Thus,

$$\nabla f = \nabla f(x_i) + H_i(x - x_i) = 0$$

If H is nonsingular,

$$x_{i+1} = x_i - H_i^{-1} \nabla f$$

Which can be shown to converge quadratically near the optimum. This method again performs better than the steepest ascent method.However, note that the method requires both the computation of second derivatives and matrix inversion at each iteration. Thus, the method is not very useful in practice for functions with large numbers of variables. Furthermore, Newtons method may not converge if the starting point is not close to the optimum.

Algorithm:

- **Input:** f : $\mathbb{R}^n \rightarrow \mathbb{R}$ a twice-differentiable function $x^{(0)}$ an initial solution
- **Output:** x^*, a local minimum of the cost function f.

1. **begin**
2. $k \leftarrow 0$;
3. **while** STOP-CRIT **and** $(k < k_{max})$ **do**
4. $x^{k+1} \leftarrow x^k + \delta^k$;
5. with $\delta^k = -(H_f(x^k))^{-1} \bigtriangledown f(x^k)$;
6. $k \leftarrow k + 1$;
7. **return** x^k
8. **end**

2.3.3 Quasi-Newton Method

Quasi-Newton, or variable metric, methods seek to estimate the direct path to the optimum in a manner similar to Newton's method. However, notice that the Hessian matrix is composed of the second derivatives of f that vary from step to step. Quasi-Newton methods attempt to avoid these difficulties by approximating H with another matrix A using only first partial derivatives of f. The approach involves starting with an initial approximation of H^{-1} and updating and improving it with each iteration. The methods are called quasi-Newton because we do not use the true Hessian, rather an approximation.

You can see the comprasion Newton's method to Quasi Newton Method. (Table 2)

Newton's Method	*Quasi-Newton Method*
Computationally expensive	Computationally cheap
Slow computation	Fast(er) computation
Need to iteratively calculate second derivative	No need for second derivative
Need to iteratively solve linear system of equations	No need to solve linear system of equations
Less convergence steps	More convergence steps
More precise convergence path	Less precise convergence path

Table 2: Comprasion Newton's Method to Quasi- Newton

2.4 Conclusions

In this paper we deal with newton type methods algorithms, its advantages and pitfalls. We proposed to use this approach if we can calculate derivatives easily. If we have second derivatives we can compute local minimum or maximum of given function. (1-D,2-D) The disadvantages of using this method are numerous. First of all, it is not guaranteed that Newton's method will converge if we select an initial root which that is too far from the exact root. Likewise, if our tangent line becomes parallel or almost parallel to the x-axis, we are not guaranteed convergence with the use of this method. Another disadvantage is that we must have a functional representation of the derivative of our function, which is not always possible if we working only from given data.

2.5 References

- Ramin S. Esfandiari,Numerical Methods for Engineers Numerical Methods for Engineers and Scientists
- M.L.N. Gonalves, J.G. Melo / Journal of Computational and Applied Mathematics 2017,
- Meza JC. A class of trust-region methods for parallel optimization. SIAM JOptim 2002,
- https://www.cs.ccu.edu.tw
- http://www.brnt.eu/
- http://www.tau.ac.il/ tsirel/dump/Static/knowino.org/wiki/
- https://011181hqf-y-https-onlinelibrary-wiley-com.proxy.gtu.deep-knowledge.net/doi/epdf/10.1
- https://web.stanford.edu/class/cme334/docs/newton-type-methods.pdf
- https://www.whitman.edu

- http://www2.lawrence.edu
- http://en.citizendium.org