Beyond Newton's method

Thomas P. Minka

August 10, 2000

Abstract

Newton's method for optimization is equivalent to iteratively maximizing a local quadratic approximation to the objective function. But some functions are not well-approximated by a quadratic, leading to slow convergence, and some have turning points where the curvature changes sign, leading to failure. The fix this, we can use a more appropriate choice of local approximation than quadratic, based on the type of function we are optimizing. This paper demonstrates two such generalized Newton rules. Like Newton's method, they only involve the first two derivatives of the function, yet give much improved convergence.

1 Newton's method

Newton's method is usually described in terms of root-finding, but it can also be understood as maximizing a local quadratic approximation to the objective function. Let the objective be $f(x)$. A quadratic approximation about x_0 has the form

$$\begin{aligned} g(x) &= k + \frac{b}{2}(x-a)^2 && (1)\\ g'(x) &= b(x-a) && (2)\\ g''(x) &= b && (3) \end{aligned}$$

With three free parameters, we can make g and its first two derivatives match those of f at x_0. That is, (k, b, a) are given by

$$\begin{aligned} k + \frac{b}{2}(x_0-a)^2 &= f(x_0) && (4)\\ b(x_0-a) &= f'(x_0) && (5)\\ b &= f''(x_0) && (6) \end{aligned}$$

whose solution is

$$\begin{aligned} b &= f''(x_0) && (7)\\ a &= x_0 - f'(x_0)/f''(x_0) && (8)\\ k &= f(x_0) - \frac{f'(x_0)^2}{2f''(x_0)} && (9) \end{aligned}$$

If $b < 0$, then the maximum of g is a, leading to the update rule

$$x^{new} = x - f'(x)/f''(x) \tag{10}$$

which is exactly Newton's method for optimization. Unfortunately, if $b = f''(x) \geq 0$, then the method fails and we must use some other optimization technique to get closer to the optimum of f.

2 A non-quadratic variation

In maximum-likelihood estimation of a Dirichlet distribution, we encounter the objective

$$f(x) = \log \frac{\Gamma(x)\exp(xs)}{\prod_{k=1}^K \Gamma(xm_k)} \qquad x > 0 \tag{11}$$

This objective is convex, but Newton's method may still fail by producing a negative x value. We can get a faster and more stable algorithm by using a local approximation of the form

$$\begin{aligned} g(x) &= k + a\log(x) + bx && (12)\\ g'(x) &= a/x + b && (13)\\ g''(x) &= -a/x^2 && (14) \end{aligned}$$

Matching f and its derivatives at x_0 requires

$$\begin{aligned} a &= -x_0^2 f''(x_0) && (15)\\ b &= f'(x_0) - a/x_0 && (16) \end{aligned}$$

Figure 1 demonstrates the high quality of this approximation compared to a quadratic.

Since $f''(x) \leq 0$ for all x, we know $a \geq 0$. If in addition $b < 0$, then the maximum of g is $-a/b$, giving the update rule

$$\frac{1}{x^{new}} = \frac{1}{x} + \frac{f'(x)}{x^2 f''(x)} \tag{17}$$

If it happens that $b \geq 0$, then some other method must be used.

Note that this update *cannot* be achieved by simply changing the parameterization of f and applying Newton's method to the new parameterization. For example, if we write $f(x) = f(1/y)$ and use a quadratic approximation in y, we get the Newton update

$$\begin{aligned} y^{new} &= y - \frac{f'(1/y)(-1/y^2)}{f''(1/y)(1/y^4) + f'(1/y)(2/y^3)} && (18)\\ \text{or } \frac{1}{x^{new}} &= \frac{1}{x} + \frac{f'(x)}{x^2 f''(x) + 2xf'(x)} && (19) \end{aligned}$$

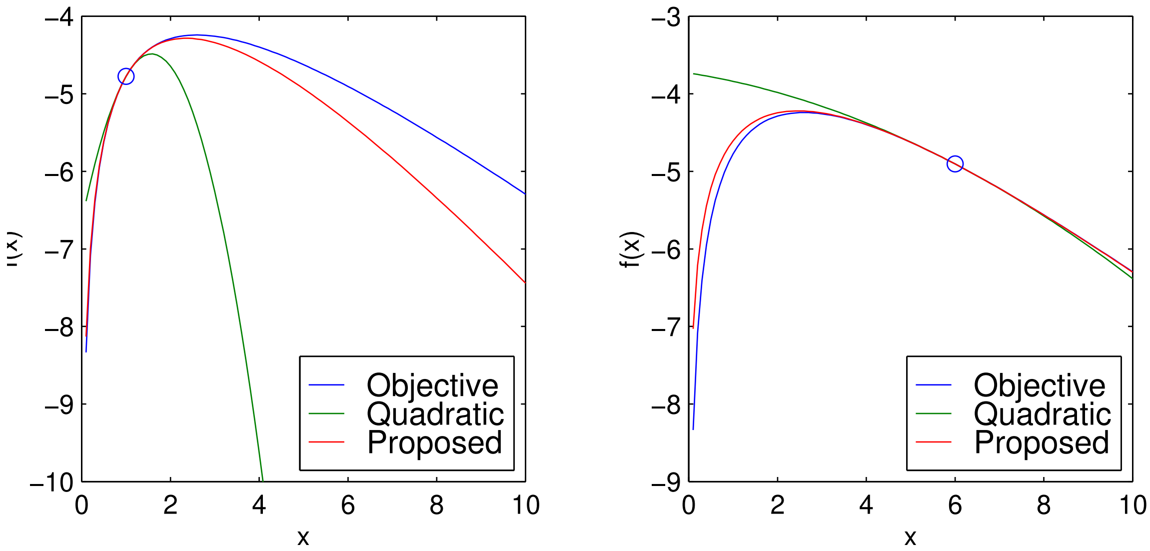

Figure 1: Approximations to the objective (11) at $x = 1$ (left) and $x = 6$ (right). Here $K = 3$, $\mathbf{m} = [1/2, 1/6, 1/3]$, and $s = -1.5$.

Iteration	Newton $y = 1/x$ (19)	Newton $y = \log(x)$ (20)	Proposed method (17)
0	$x = 6$	$x = 6$	$x = 6$
1	4.37629402002204	3.32635155274492	2.46069962590123
2	3.37583855493481	2.65619730648832	2.59239008053976
3	2.83325972215703	2.59371556140329	2.59311961991097
4	2.62645752464592	2.59311969529076	2.59311964068914
5	2.59390440938786	2.59311964068914	2.59311964068914

Figure 2: Convergence of three iteration schemes starting from $x = 6$ on the objective in figure 1. The proposed method reaches machine precision in 4 iterations.

which is as close as you can get to (17) using reparameterization.

Since x must be positive, it is also tempting to reparameterize with $f(x) = f(e^y)$. Applying Newton's method to y gives

$$x^{new} = x \exp\left(-\frac{f'(x)}{xf''(x) + f'(x)}\right) \tag{20}$$

Figure 2 compares the convergence rate of the three updates (17), (19), and (20).

3 Another variation

In maximum-likelihood estimation of certain Gaussian models, we encounter the objective

$$f(x) = -\sum_{i=1}^{n} \log(x + v_i) - \frac{s_i}{x + v_i} \qquad x \geq 0 \tag{21}$$

This objective is not convex and has turning points which confound Newton's method. Instead of quadratic g, consider a local approximation of the form

$$\begin{aligned} g(x) &= k - n\log(x + a) - \frac{b}{x + a} && (22) \\ g'(x) &= -\frac{n}{x + a} + \frac{b}{(x + a)^2} && (23) \\ g''(x) &= \frac{n}{(x + a)^2} - \frac{2b}{(x + a)^3} && (24) \end{aligned}$$

Matching f and its derivatives at x_0 requires

$$\begin{aligned} b &= n(x_0 + a) + f'(x_0)(x_0 + a)^2 && (25) \\ a &= \begin{cases} -\frac{\sqrt{f'(x_0)^2 - nf''(x_0)} + f'(x_0)}{f''(x_0)} - x_0 & \text{if } f''(x_0) \neq 0 \\ -\frac{n}{2f'(x_0)} - x_0 & \text{if } f''(x_0) = 0 \end{cases} && (26) \end{aligned}$$

Figure 3 demonstrates the high quality of this approximation compared to a quadratic.

If $a \geq 0$ and $b > 0$, the maximum of g is $b/n - a$, so the update rule is

$$x^{new} = x + \frac{f'(x)}{n}(x + a)^2 \tag{27}$$

If the conditions aren't met, which can happen when we are far from the maximum, then some other scheme must be used, just as with Newton's method. However, the method does *not* fail when $f''(x)$ crosses zero. Figure 4 compares the convergence rate of the updates (10), (20), and (27).

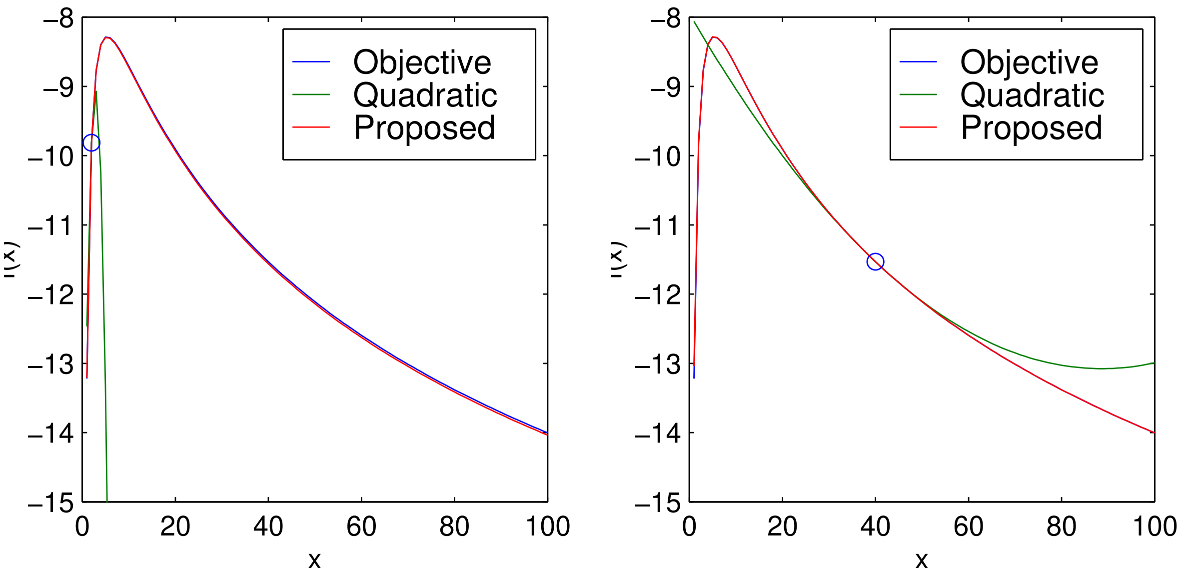

Figure 3: Approximations to the objective (21) at $x = 2$ (left) and $x = 40$ (right). Here $n = 3$ and $\mathbf{v} = [1, 1/2, 1/3]$, $\mathbf{s} = [2, 5, 10]$. In both cases, the proposed approximation (22) is nearly identical to the true objective.

Iteration	Newton (10)	Newton $y = \log(x)$ (20)	Proposed method (27)
0	$x = 2$	$x = 2$	$x = 2$
1	2.88957666375356	4.45604707043312	5.33228321620505
2	3.88206057335333	5.30261828233160	5.36035707890124
3	4.75937270469894	5.36013962130173	5.36035751320994
4	5.24746847271009	5.36035751012362	5.36035751320994

Figure 4: Convergence of three iteration schemes starting from $x = 2$ on the objective in figure 3. Update (19) fails and is not shown. The proposed method reaches machine precision in 3 iterations. Starting from $x = 40$, the proposed method converges equally fast, while the updates (10) and (20) fail.