Newton's Method

Newton's method is a classical technique for finding the root of a general differentiable function $f(x) : \mathbb{R} \to \mathbb{R}$. That is, we want to find an $x \in \mathbb{R}$ such that

$$f(x) = 0.$$

As you probably learned in high school, one technique for doing this is to start at some guess x_0, and then follow the iteration

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

You can draw your own picture here:

Of course, there can be many roots, and which one we converge to will depend on what we choose for x_0. It is also very much possible that the iterations do not converge for certain (or even almost all) initial values x_0. But there is a classical convergence theory that says that once we are close enough to a particular root x_0, we will have

$$\underbrace{|x_0 - x^{(k+1)}|}_{\epsilon_{k+1}} \leq C \cdot \underbrace{(x_0 - x^{(k)})^2}_{\epsilon_k^2},$$

where the constant C depends on the ratio between the first and second derivatives in the interval[1]. around the root x_0:

$$C = \sup_{x \in \mathcal{I}} \frac{|f''(x)|}{2|f'(x)|}.$$

The take-away here is that close to the solution, Newton's methods exhibits *quadratic convergence*: the error at the next iteration is proportional to the square of the error at the last iteration. Since we are concerned with ϵ_k small, $\epsilon_k \ll 1$, this means that under the right conditions, the error goes down in dramatic fashion from iteration to iteration.

Notice that applying the technique requires that f is differentiable, but the convergence guarantee depends on f be twice (continuously) differentiable.

When $f(x)$ is convex, twice differentiable, and has a minimizer, we can find a minimizer by applying Newton's method to the derivative. We start at some initial guess x_0, and then take

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}. \tag{1}$$

Again, if f is three-times continuously differentiable, we converge to the global minimizer quadratically with a constant that depends on

$$C = \sup_{x \in \mathcal{I}} \frac{1}{2} \frac{|f'''(x)|}{|f''(x)|},$$

[1]There are various technical conditions that f must obey on $\mathcal{I}$ for this result to hold, including the second derivative being continuous and the first derivative not being equal to zero. Also, the condition "close enough" is characterized by looking at ratios of derivatives at the root and on $\mathcal{I}$. The Wikipedia article on this is not bad: https://en.wikipedia.org/wiki/Newton's_method

for an appropriate interval $\mathcal{I}$ around the solution. Again, applying the method relies on us being able to compute first and second derivatives of f, and the analysis relies on f being three-times differentiable.

We can interpret the iteration (1) above in the following way:

1. At $x^{(k)}$, approximate $f(x)$ using the Taylor expansion

$$f(x) \approx f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) + \frac{1}{2} f''(x^{(k)})(x - x^{(k)})^2.$$

2. Find the exact minimizer of this quadratic approximation. Taking the derivative of the expansion above and setting it equal to zero yields the following optimality condition for $\hat{x}$ to be a minimizer:

$$(\hat{x} - x^{(k)}) f''(x^{(k)}) = -f'(x^{(k)}).$$

   This is just a re-arrangement of the iteration (1).

3. Take $x^{(k+1)} = \hat{x}$.

This last interpretation extends naturally to the case where $f(\boldsymbol{x})$ is a function of many variables, $f : \mathbb{R}^N \to \mathbb{R}$. We know that if f is convex and twice differentiable, we have a minimizer $\boldsymbol{x}^\star$ when $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$. Newton's method to find such a minimizer proceeds as above. We start with an initial guess $\boldsymbol{x}^{(0)}$, and use the following iteration:

1. Take a Taylor approximation around $f(\boldsymbol{x}^{(k)})$:

$$f(\boldsymbol{x}) \approx f(\boldsymbol{x}^{(k)}) + \langle \boldsymbol{x} - \boldsymbol{x}^{(k)}, \nabla f(\boldsymbol{x}^{(k)}) \rangle + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}^{(k)})^{\mathrm{T}} \nabla^2 f(\boldsymbol{x}^{(k)}) (\boldsymbol{x} - \boldsymbol{x}^{(k)})$$

2. Find the exact minimizer $\hat{\boldsymbol{x}}$ to this approximation. This gives

us the problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}}\, \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}^{(k)}) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^{(k)})^{\mathrm{T}}\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}^{(k)}),$$

where

$$\begin{aligned}\boldsymbol{H} &= \nabla^2 f(\boldsymbol{x}^{(k)}) = N\times N \text{ Hessian matrix at } \boldsymbol{x}^{(k)}\\ \boldsymbol{g} &= \nabla f(\boldsymbol{x}^{(k)}) = N\times 1 \text{ gradient vector at } \boldsymbol{x}^{(k)}.\end{aligned}$$

Since $\boldsymbol{H}$ is symmetric and positive semi-definite, we know that the conditions for $\hat{\boldsymbol{x}}$ being a minimizer[2] are

$$\boldsymbol{H}(\boldsymbol{x}-\boldsymbol{x}^{(k)}) = -\boldsymbol{g}.$$

If $\boldsymbol{H}$ is invertible (sym+def), then we have a unique minimizer and

$$\hat{\boldsymbol{x}} = \boldsymbol{x}^{(k)} - \boldsymbol{H}^{-1}\boldsymbol{g}.$$

3. Take $\boldsymbol{x}^{(k+1)} = \hat{\boldsymbol{x}}$.

This procedure is often referred to as a *pure Newton step*, as it does not involve the selection of a step size. In practice, however, it is often beneficial to choose the step direction as

$$\boldsymbol{d}^{(k)} = -\left(\nabla^2 f(\boldsymbol{x}^{(k)})\right)^{-1}\nabla f(\boldsymbol{x}^{(k)}),$$

and then choose a step size t_k using a backtracking line search, and then take

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + t_k\boldsymbol{d}^{(k)}$$

as before.

[2] Take the gradient of this new expression and set it equal to $\mathbf{0}$.

Convergence of Newton's Method

Suppose that $f(\boldsymbol{x})$ is strongly convex,

$$m\mathbf{I} \preceq \nabla^2 f(\boldsymbol{x}) \preceq M\mathbf{I}, \quad \forall \boldsymbol{x} \in \mathbb{R}^N,$$

and that its Hessian is Lipschitz,

$$\|\nabla^2 f(\boldsymbol{x}) - \nabla^2 f(\boldsymbol{y})\| \ \leq \ L\|\boldsymbol{x} - \boldsymbol{y}\|_2.$$

(The norm on the left-hand side above is the standard operator norm.) We will show that the Newton algorithm coupled with an exact line search[3] converges to precision ϵ:

$$f(\boldsymbol{x}^{(k)}) - p^\star \leq \epsilon,$$

for a number of iterations

$$k \ \geq \ C_1 \left(f(\boldsymbol{x}^{(0)}) - p^\star \right) + \log_2 \log_2(\epsilon_0/\epsilon),$$

where we can take the constants above to be $C_1 = M^2L^2/m^5$ and $\epsilon_0 = 2m^3/L^2$. Qualitatively, this says that Newton's method takes a constant number of iterations to converge to any reasonable precision — we can bound $\log_2 \log_2(\epsilon_0/\epsilon) \leq 6$ (say) for ridiculously small values of ϵ.

To establish this result, we break the analysis into two stages. In the first, the *damped Newton stage*, we are far from the solution (as measured by $\|\nabla f(\boldsymbol{x}^{(k)})\|_2$), but we make constant progress towards the answer. Specifically, we will show that in this stage,

$$f(\boldsymbol{x}^{(k+1)}) - f(\boldsymbol{x}^{(k)}) \ \leq \ 1/C_1.$$

[3] These results are easily extended to backtracking line searches; we are just using an exact line search to make the exposition easier. See [BV04, Sec. 9.5.3] for the analysis with backtracking.

It is clear, then, that the number of damped Newton steps is no greater than $C_1 \left(f(\boldsymbol{x}^{(0)}) - p^\star\right)$.

We will then show that when $\|\nabla f(\boldsymbol{x}^{(k)}\|_2$ is small enough, the gap closes dramatically at every iteration. We call this the *quadratic convergence stage*, as we will be able to show that once the algorithm enters this stage at iteration ℓ, for all $k > \ell$,

$$\|\nabla f(\boldsymbol{x}^{(k)})\|_2 \;\leq\; C_2 \cdot 2^{-2^{k-\ell}},$$

where $C_2 = L/(2m^2)$ is another constant.

Damped phase: We are in this stage when

$$\|\nabla f(\boldsymbol{x}^{(k)})\|_2 \geq m^2/L.$$

We take $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + t_{\text{exact}}\boldsymbol{d}^{(k+1)}$, where

$$\boldsymbol{d}^{(k+1)} = -\nabla^2 f(\boldsymbol{x}^{(k)})^{-1}\nabla f(\boldsymbol{x}^{(k)}),$$

and t_{exact} is the result of an exact line search[4]:

$$t_{\text{exact}} = \arg\min_{0\leq t\leq 1} f(\boldsymbol{x}^{(k)} + t\boldsymbol{d}^{(k+1)}).$$

With the current Newton decrement denoted as

$$\lambda_k^2 = -\nabla f(\boldsymbol{x}^{(k)})^{\mathrm{T}}\boldsymbol{d}^{(k+1)} = \|\boldsymbol{d}^{(k+1)}\|_2^2,$$

we know that

$$\begin{aligned} f(\boldsymbol{x}^{(k)} + t\boldsymbol{d}^{(k+1)}) &\leq f(\boldsymbol{x}^{(k)}) - t\lambda_k^2 + \frac{M}{2}\|t\boldsymbol{d}^{(k+1)}\|_2^2 \\ &\leq f(\boldsymbol{x}^{(k)}) - t\lambda_k^2 + \frac{M}{2m}t^2\lambda_k^2, \end{aligned}$$

[4]For convenience, we are not letting t be larger than 1, just as in a backtracking method.

where the second step follows from the fact that the largest eigenvalue of $[\nabla^2 f(\boldsymbol{x}^{(k)})]^{-1}$ is at most $1/m$. Plugging in $t = m/M$ above yields

$$\begin{aligned} f(\boldsymbol{x}^{(k)} + t_{\text{exact}}\boldsymbol{d}^{(k+1)}) - f(\boldsymbol{x}^{(k)}) &\leq -\frac{m}{M}\lambda_k^2 \\ &\leq -\frac{m}{M^2}\|\nabla f(\boldsymbol{x}^{(k)})\|_2^2 \\ &\leq -\frac{m^5}{L^2M^2}. \end{aligned}$$

Quadratic convergence: When

$$\|\nabla f(\boldsymbol{x}^{(k)})\|_2 < m^2/L,$$

we start to settle things very quickly. We will assume that in this stage, we choose the step size to be $t = 1$. In fact, you can show that under very mild assumptions on the backtracking parameter ($\alpha < 1/3$, to be specific), backtracking will indeed not backtrack at all and return $t = 1$ (see [BV04, p. 490]).

We start by pointing out that by construction,

$$\nabla^2 f(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k+1)} = -\nabla f(\boldsymbol{x}^{(k)}),$$

and so by the Taylor theorem

$$\begin{aligned} \nabla f(\boldsymbol{x}^{(k+1)}) &= \nabla f(\boldsymbol{x}^{(k)} + \boldsymbol{d}^{(k+1)}) - \nabla f(\boldsymbol{x}^{(k)}) - \nabla^2 f(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k+1)} \\ &= \int_0^1 \nabla^2 f(\boldsymbol{x}^{(k)} + t\boldsymbol{d}^{(k+1)})\boldsymbol{d}^{(k+1)}\,\mathrm{d}t - \nabla^2 f(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k+1)} \\ &= \int_0^1 \left[\nabla^2 f(\boldsymbol{x}^{(k)} + t\boldsymbol{d}^{(k+1)}) - \nabla^2 f(\boldsymbol{x}^{(k)})\right]\boldsymbol{d}^{(k+1)}\,\mathrm{d}t. \end{aligned}$$

Thus

$$\begin{aligned}\|\nabla f(\boldsymbol{x}^{(k+1)})\|_2 &\leq \int_0^1 \|\nabla^2 f(\boldsymbol{x}^{(k)} + t\boldsymbol{d}^{(k+1)}) - \nabla^2 f(\boldsymbol{x}^{(k)})\| \cdot \|\boldsymbol{d}^{(k+1)}\|_2 \, \mathrm{d}t \\ &\leq \int_0^1 t^2 L \|\boldsymbol{d}^{(k+1)}\|_2^2 \, \mathrm{d}t \\ &= \frac{L}{2} \|\nabla f(\boldsymbol{x}^{(k)})^{-1} \nabla f(\boldsymbol{x}^{(k)})\|_2^2 \\ &\leq \frac{L}{2m^2} \|\nabla f(\boldsymbol{x}^{(k)})\|_2^2.\end{aligned}$$

Since $\|\nabla f(\boldsymbol{x}^{(k)})\|_2 \leq m^2/L$, we have

$$\frac{L}{2m^2}\|\nabla f(\boldsymbol{x}^{(k+1)})\|_2 \;\leq\; \left(\frac{L}{2m^2}\|\nabla f(\boldsymbol{x}^{(k)})\|_2\right)^2 \;\leq\; \left(\frac{1}{2}\right)^2.$$

That is, at every iteration, we are **squaring** the error (which is less than $1/2$). If we entered this stage at iteration ℓ, this means

$$\frac{L}{2m^2}\|\nabla f(\boldsymbol{x}^{(k)})\|_2 \;\leq\; \left(\frac{L}{2m^2}\|\nabla f(\boldsymbol{x}^{(\ell)})\|_2\right)^{2^{k-\ell}} \;\leq\; \left(\frac{1}{2}\right)^{2^{k-\ell}}.$$

Then using the strong convexity of f,

$$f(\boldsymbol{x}^{(k)}) - p^\star \;\leq\; \frac{1}{2m}\|\nabla f(\boldsymbol{x}^{(k)})\|_2^2 \;\leq\; \frac{2m^3}{L^2}\left(\frac{1}{2}\right)^{2^{k-\ell+1}}.$$

The right hand side above is less than ϵ when

$$k - \ell + 1 \geq \log_2 \log_2(\epsilon_0/\epsilon), \quad \epsilon_0 = 2m^3/L^2,$$

so we spend no more than $\log_2 \log_2(\epsilon_0/\epsilon)$ iterations in this phase.

Note that

$$\epsilon = 10^{-20}\epsilon_0 \quad \Rightarrow \quad \log_2 \log_2(\epsilon_0/\epsilon) = 6.0539.$$

Convergence criteria: the Newton decrement

We know that at the minimia of a smooth convex functional we will have $\nabla f(\boldsymbol{x}) = \mathbf{0}$. So a natural test for convergence is to measure how far away $\nabla f(\boldsymbol{x})$ is from $\mathbf{0}$; that is, we say we are converged when the norm of $\nabla f(\boldsymbol{x})$ is below some threshold (call it ϵ):

$$\text{stop when } \|\nabla f(\boldsymbol{x}^{(k)})\| \leq \epsilon.$$

Which norm should we use?

The natural instinct here is to go with the standard Euclidean (ℓ_2) norm, stopping when

$$\|\nabla f(\boldsymbol{x}^{(k)})\|_2 \leq \epsilon,$$

and in fact, this quantity played a key role in our analysis above. But there is something that is unsatisfying about using the Euclidean norm, and this problem also extends to the way we approached the analysis in the previous section. An interesting feature of Newton's method is that it is *affine invariant*; if we simply change the coordinates, the iterates change accordingly. For example, let $\boldsymbol{T}$ be a $N \times N$ invertible matrix, and set $\tilde{f}(\boldsymbol{x}) = f(\boldsymbol{T}\boldsymbol{x})$. Suppose we run Newton's method to try to find a minima of f starting at $\boldsymbol{x}^{(0)}$ and computing iterates $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots$. Then we run Newton's method on $\tilde{f}$ starting at $\boldsymbol{T}^{-1}\boldsymbol{x}^{(0)}$ and compute iterates $\tilde{\boldsymbol{x}}^{(1)}, \tilde{\boldsymbol{x}}^{(2)}, \ldots$. This second set of iterates will follow the same progression as the first under transformation by $\boldsymbol{T}$:

$$\tilde{\boldsymbol{x}}^{(k)} = \boldsymbol{T}^{-1}\boldsymbol{x}^{(k)}, \quad k = 1, 2, \ldots$$

The problem, then, with the the Euclidean norm of the gradient is that it is not affinely invariant:

$$\|\nabla \tilde{f}(\boldsymbol{x})\|_2 \neq \|\nabla f(\boldsymbol{T}\boldsymbol{x})\|_2 \quad \text{for general } \boldsymbol{T}.$$

(Apply the chain rule.)

A criteria that is affinely invariant is the Newton decrement:

$$\lambda(\boldsymbol{x}) = \sqrt{\boldsymbol{g}^{\mathrm{T}}\boldsymbol{H}^{-1}\boldsymbol{g}}, \quad \boldsymbol{g} = \nabla f(\boldsymbol{x}), \quad \boldsymbol{H} = \nabla^2 f(\boldsymbol{x}).$$

(Again, you can work this out with a little effort by applying the chain rule.) These are various ways you can interpret this: one is as size of the gradient in the norm induced by $\boldsymbol{H}^{-1}$:

$$\lambda(\boldsymbol{x}) = \|\nabla f(x)\|_{\boldsymbol{H}^{-1}}.$$

Of course, the norm itself depends on the point $\boldsymbol{x}$. You can also think of it as the directional derivative in the direction we are taking a Newton step; if $\boldsymbol{d} = -(\nabla^2 f(\boldsymbol{x}))^{-1}\nabla f(\boldsymbol{x})$, then

$$\langle \boldsymbol{d}, \nabla f(\boldsymbol{x}) \rangle = -\lambda(\boldsymbol{x})^2.$$

At any rate, the convergence criteria for Newton's method is usually whether $\lambda(\boldsymbol{x}^{(k)})$ is below some threshold.

Self-concordant functions

There is an alternative analysis of Newton's method that is more satisfying in that it gives an affinely invariant bound, and it does not depend on the constants m, M, L that are usually unknown. The analysis holds for functions that are self concordant, a term that we define below.

Definition. We say that a convex function of one variable $f : \mathbb{R} \to \mathbb{R}$ is *self-concordant* if

$$|f'''(x)| \leq 2f''(x)^{3/2}, \quad \text{for all} \quad x \in \operatorname{dom} f.$$

We say that a convex function of multiple variables $f : \mathbb{R}^N \to \mathbb{R}$ is self-concordant if

$$g(t) = f(\boldsymbol{x} + t\boldsymbol{v}) \text{ is self-concordant for all } \boldsymbol{x} \in \operatorname{dom} f, \ \boldsymbol{v} \in \mathbb{R}^N.$$

We should note that the constant 2 that appears in front of the $f''(x)$ above is somewhat arbitrary — if there is any uniform bound on the ratio of $|f'''(x)|$ to $f''(x)^{3/2}$, then f can be made self-concordant simply by re-scaling.

We mention a few important examples (see [BV04, Chapter 9.6] for many more).

- Since the third derivative of all linear and quadratic functionals is zero, they are self-concordant.
- $f(x) = -\log(x)$ is self-concordant
- $f(\boldsymbol{X}) = -\log\det \boldsymbol{X}$ for $\boldsymbol{X} \in S^N_{++}$ is self-concordant
- Self-concordance is preserved under composition with an affine

transformation, so for example

$$f(\boldsymbol{x}) = -\sum_{m=1}^{M} \log(b_m - \boldsymbol{a}_m^{\mathrm{T}}\boldsymbol{x}) \quad \text{on } \{\boldsymbol{x} : \boldsymbol{a}_m^{\mathrm{T}} \leq b_m,\ m = 1, \ldots, M\}$$

is self-concordant. Functions of the above form will play a major role when we talk about log-barrier methods for contrained optimization.

Using a line of argumentation not too different than in the classical analysis in the last section, we have the following result for the convergence of Newton's method (again, see [BV04, Chapter 9] for the details):

If $f(\boldsymbol{x}) : \mathbb{R}^N \to \mathbb{R}$ is self-concordant, then Newton iterations starting from $\boldsymbol{x}^{(0)}$ coupled with standard backtracking line search will have

$$f(\boldsymbol{x}^{(k)}) - -p^\star \leq \epsilon$$

when

$$k \geq C\epsilon_0 + \log_2 \log_2(1/\epsilon), \quad \epsilon_0 = f(\boldsymbol{x}^{(0)}) - -p^\star.$$

The constant C above depends only on the backtracking parameters.

Again, we will fully appreciate this result when we talk about log barrier techniques a little later.

References

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.