Minimization Techniques

Larry Fenn

Outline of Notes

Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$, the task is to find the extrema (from here on called $\vec{c}$) and roots of f (from here on called $\vec{\alpha}$). In practice, a robust technique for finding minima is enough: to find a maxima, we reduce the problem to finding a minima of $-f$, and for finding roots, we reduce the problem to finding minima of $\langle f, f\rangle$ or f^2. These notes detail two techniques: gradient descent, and Newton's method.

Contents

1 Preliminary

For functions $f\colon \mathbb{R}^n \to \mathbb{R}$ the notion of a root or extrema is straightforward: making use of the default ordering of real numbers in the $\mathbb{R}$, we have the familiar definitions of maxima, minima, and roots.

1. $\vec{c}$ is a global/local maximum of f if $\forall \vec{x} \in \mathbb{R}^n / \exists D \subset \mathbb{R}^n, \forall \vec{x} \in D\colon f(\vec{x}) \leq f(\vec{c})$.
2. $\vec{c}$ is a global/local minimum of f if $\forall \vec{x} \in \mathbb{R}^n / \exists D \subset \mathbb{R}^n, \forall \vec{x} \in D\colon f(\vec{x}) \geq f(\vec{c})$.
3. $\vec{\alpha}$ is a root of f if $f(\vec{\alpha}) = 0$.

If we had a technique that could only compute the minima $\vec{c}$ of a function f, then note that for the function $-f$ we have that $\forall \vec{x}\colon f(\vec{x}) \geq f(\vec{c})$; thus $\forall \vec{x}\colon -f(\vec{x}) \leq -f(\vec{c})$. In other words, if $\vec{c}$ is a minima of f, then it is also a maxima of $-f$ (and vice versa). This indicates that it is sufficient for our interests to only consider seeking the minima of functions.

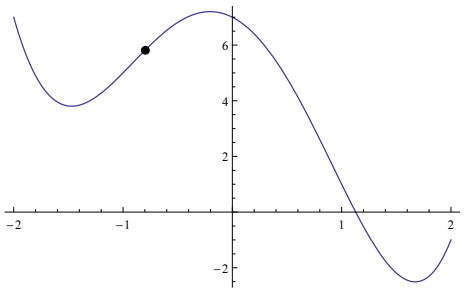
Similarly, if we were seeking a root finding technique for $f\colon \mathbb{R}^n \to \mathbb{R}$, recognize that if f has a root, $\vec{\alpha}$, then the function $f(\vec{x})^2 \geq 0$ must have a minima at $\vec{\alpha}$; $f(\vec{\alpha})^2 = 0$. By making use of this property we see that any technique for giving us a minima of an arbitrary function can be employed to find roots. The minima of $f(\vec{x})^2$ must be the roots of $f(\vec{x})$.

For more general, vector-valued functions $f\colon \mathbb{R}^n \to \mathbb{R}^m$, we can determine when $f(\vec{x}) = \vec{0}$ by using the same procedure. In this instance, the Euclidean norm in the range (or really, any norm we define) $\|\cdot\|_2\colon \mathbb{R}^m \to \mathbb{R}^+$ allows us to apply the previous techniques to $\|f(\vec{x})\|_2\colon \mathbb{R}^n \to \mathbb{R}^+ \cup \{0\}$ for finding roots and extrema. In general, if there is an inner product then we can take the induced norm and apply it to the function $\langle f, f\rangle$ to arrive at a similar conclusion. For the rest of this document, we will address only functions that map into $\mathbb{R}$.

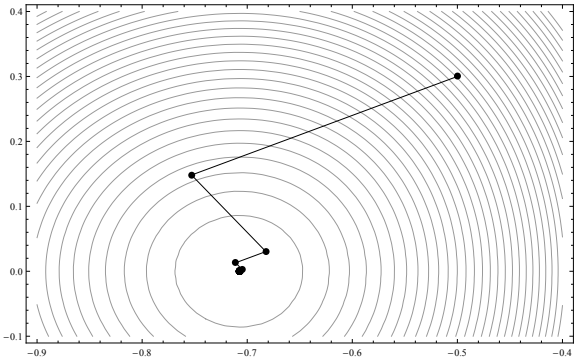
For the following techniques, we will be using the concept of a *line search*. The general idea is that we will parametrize a line that contains our initial guess. That parametrized line will allow us to use single variable calculus: given $f\colon \mathbb{R}^n \to \mathbb{R}$ and the parametrized line $\gamma\colon \mathbb{R} \to \mathbb{R}^n$ we can consider $f \circ \gamma(t)$ as a map from $\mathbb{R} \to \mathbb{R}$. The extrema we find for $f \circ \gamma$ will ideally be close to extrema for f overall. The key, and difference between these methods, is hence how they go about determining the right line to travel on. In essence, we are seeking a direction of travel from our starting point that places us closer to the extrema.

2 Gradient Descent Line Search

Figure 1



(a) $f(x) = x^4 - 5x^2 - 3x + 7$



(b) Gradient descent line search seeking the minima of $f(x, y) = xe^{-(x^2+y^2)}$

2.1 Motivation

Imagine climbing a hill. The shortest path to the top of the hill is exactly that path that follows the steepest slope. If we could only stare at our feet as we climbed a hill, it would still be possible to make our way to the summit in a direct fashion: gauge which step makes the most uphill progress, and take it. Likewise, we could find our way to the deepest point of a ravine by using the same idea. In fact, water always obeys this principle: for a river flowing from glacier to ocean, each drop of water (as a consequence of the laws of physics) is traveling by this pattern. In more abstract terms, if we could establish a local gauge of "steepness", then we could find our way to the maxima or minima by traveling along the steepest path. Before we discuss the more general multivariable case, let's consider the single dimensional case from traditional calculus.

2.2 One Dimensional Case

Consider the graphed polynomial above, and let our starting point be marked by the dot. If we compute the first derivative at our point, we will get a single number that reflects the current slope at the given point. In this instance, we will get some positive quantity; an indication that if (for a sufficiently small step size) we step to the right of the current position, the function value will increase. In other words, the first derivative being positive at the point directly implies that the closest extrema to the left of our starting point must be a minima, and the closest extrema to the right must be a maxima. It's the equivalent of, upon finding yourself on a hill, knowing which direction is "uphill" and which is "downhill". Thus, if we use a one-dimensional root finding technique to find the closest root of f' to the left of our initial guess, that root must correspond to a minima of the function value.

In summary, given an initial guess x_0, the procedure is as follows:

1. Determine the direction of travel:
   (a) If $f'(x_0) > 0$, then travel to the left (i.e. in the negative direction); search only $x < x_0$.
   (b) If $f'(x_0) < 0$, then travel to the right (i.e. in the positive direction); search only $x > x_0$.
2. Travel along the direction of travel until $f'(x) = 0$ along the direction of travel. Assert this is the minima (or that the initial guess was bad).

This approach is not without its problems. Suppose we were trying to find the global minima of the function in figure 1a from this starting point. Following the first derivative as an indication of which direction to apply a root finding technique, and then computing the root of the first derivative to the left of our starting point, we will have found the local minima between -1 and -2. However, this is plainly not the global minima of the function. There are many ways to overcome this difficulty, however they are not presented here. Additionally, depending on the root finding technique used we may not actually know whether or not the root we find to the left of our initial guess is the closest root to our initial guess; for a pathological function like $\dfrac{\sin(x)}{x}$ we will attain an astonishing amount of incorrect answers. Without any other assistance, this means this algorithm may produce incorrect answers for certain domains of starting guesses.

One may rightly ask why we have concocted such a fuzzy and roundabout way for finding the minima of a function. Wouldn't it be simpler to use root finding techniques on the first derivative, and then evaluating the function at those points to determine where the maxima and minima are? The reason is twofold: one, we would like our algorithm to rely at most upon only "local" data- that is to say, we want to limit the scope of what we need to compute or determine (or have to know *a priori*) to just what is in the neighborhood of our guesses. This means we can compute derivatives, but we may not necessarily be able to compute all the roots of the derivative function. Our root finding techniques should thus also be local, or in any case resemble Newton's method for finding roots and only depend on local measurements. The second reason is more to due with dimensionality: while it is true in one dimension that the derivative being zero implies an extrema (or additionally, that the list of roots for the first derivative is "easy to work with"), in multidimensional contexts it can be the case that entire regions have zero gradient (the higher dimensional analogue of "steepness"). This means that any technique predicated on obtaining all the points where the slope is zero will become totally unusable at higher dimensions.

2.3 Two Dimensional Case

For a function $f\colon \mathbb{R}^2 \to \mathbb{R}$ we see that this function assigns to every point (x, y) a certain value. There are several physical models that have this type of underlying function; for imagination's sake, you can think about this in any number of ways including but not limited to electrostatic attraction, heat, density. Alternatively, an entirely geometrical treatment of the problem is feasible; associate with the function the surface given by the points $(x, y, f(x, y))$. With the geometrical treatment, we can imagine finding a minima as trying to walk our way to the lowest point on the surface. In this context, the notion of "direction of travel" is given plainly by the gradient $\nabla f(x, y)$. Recall from multivariable calculus that the gradient of a function $f\colon \mathbb{R}^2 \to \mathbb{R}$ is a vector that points in the direction of the greatest increase of the function.

Thus, for finding a minima, we want to use $-\nabla f(x, y)$, the direction of the greatest decrease. So, our extension of the technique above given a starting guess $\vec{x}_0 = (x_0, y_0)$ is as follows:

1. Determine the direction of travel from x_0: $-\nabla f(\vec{x}_0)$.
2. Travel from x_0 along the direction of travel using the following parametrized curve: $\gamma \colon \mathbb{R} \to \mathbb{R}^2 \colon \gamma(t) = \vec{x}_0 - t\nabla f(\vec{x}_0)$.
3. When $f'(\gamma(t)) = 0$, assert that this is the minima.

There is one issue with this direct extension to higher dimension. If we imagine the problem as attempting to traverse down to the lowest point on a surface, this is what we will be doing:

1. Looking at the ground, find the current steepest direction.
2. Walk in that direction until (in that direction) the ground is flat.
3. Assert that we are standing at the bottom of the surface.

The issue is that, for example, the surface on either side of the direction we're walking could continue to descend. For example, if our surface is locally saddle-shaped, you can see that this process fails to deliver the right answer. The natural fix is to have this be an *iterative* method: when we stop, we should reassess where we stand and look for a new direction of travel based on our current position.

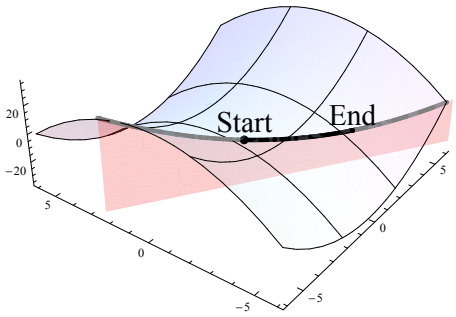


Figure 2: $\vec{x}_0 = (-1.5, -1)$
$f(x, y) = x^2 - y^2$
The traversal $f \circ \gamma$ is from the starting point to the end of the black path. The function has zero derivative exactly at the end point. The next iteration should send us on a path down the saddle, toward the viewer.

As with any iterative method, we need exit conditions. For our purposes, there are three:

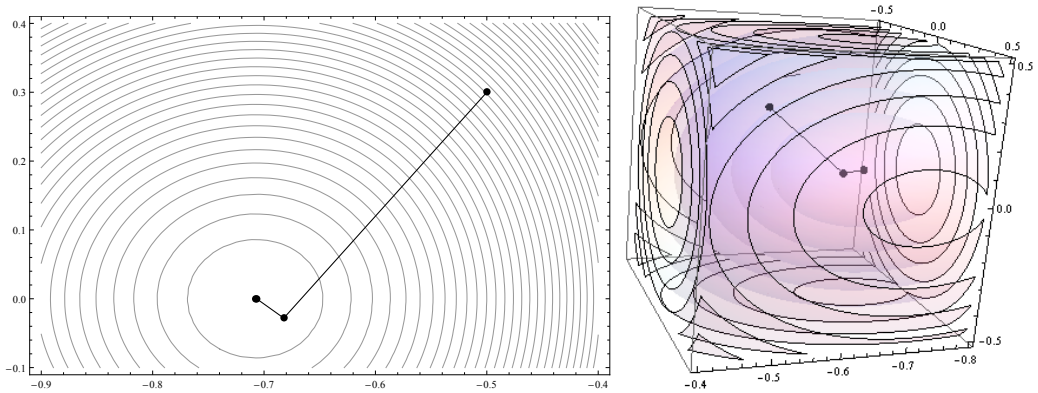
1. If we exceed an in-built iteration limit (computation time limit).
2. If our change between guesses is below a threshold, dubbed *nearness* (desired precision limit).
3. Starting at $\vec{x}_0$ if our next guess $\vec{x}_1$ has $f(\vec{x}_1) > f(\vec{x}_0)$ (bad guess).

If none of these exit conditions are met, we iterate again with a new starting point.

2.4 Multidimensional Case

This technique easily generalizes to higher dimensions. The gradient can be generalized to arbitrary dimensions; thus, while it is hard to visualize the notion of walking down hills, there is no reason that the abstract principles behind walking down hills do not apply in higher dimensions.

3 Newton's Method Line Search



(a) $f(x,y) = xe^{-(x^2+y^2)}$ (b) $f(x,y,z) = xe^{-(x^2+y^2+z^2)}$

Figure 3: Newton's method line search in 2 and 3 dimensions

3.1 Motivation

Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$, the second order Taylor expansion of f around $\vec{x}_0$ is an approximation of $f(\vec{x}_0 + \vec{s})$. Thus, if we choose $\vec{s}$ specifically to satisfy the condition that $f(\vec{x}_0 + \vec{s}) = f(\vec{x}_0)$, by Rolle's theorem we have that there must be an extrema of f located on the path from $\vec{x}_0$ to $\vec{x}_0 + \vec{s}$. In other words, we use $\vec{s}$ as our direction of travel from $\vec{x}_0$ parametrized as $\gamma(t)$, and use traditional root finding on $f'(\gamma(t))$ to determine the extrema.

3.2 One Dimensional Case

In one dimension this is a direct implication of the second derivative test from calculus. If a function is "concave up" from a to b, then the extrema (if any) that lies between a and b must be a minima. Hence if we find ourselves in a region that is concave up, we have a stronger guarantee than in the gradient descent case that traveling "downhill" will take us to a minima.

If $f\colon \mathbb{R} \to \mathbb{R}$ is twice-differentiable then its Taylor expansion about x_0 is given by

$$f(x_0 + \Delta x) = f(x_0) + f'(x_0)\Delta x + \frac{1}{2!}f''(x_0)(\Delta x)^2$$

Assume that Δx satisfies $f(x_0 + \Delta x) = f(x_0)$; thus, by Rolle's theorem there must be c between x_0 and $x_0 + \Delta x$ such that $f'(c) = 0$. Additionally, since $f(x_0 + \Delta x) = f(x_0)$ we have from the Taylor expansion that

$$0 = f'(x_0)\Delta x + \frac{1}{2!}f''(x_0)(\Delta x)^2$$

Rearranging terms we have that $\Delta x = -2\dfrac{f'(x_0)}{f''(x_0)}$. This tells us which direction we should travel in; in one dimension, this amounts to left or right (negative or positive).

By what we have shown, the point c must lie along this direction of travel, and the extrema must therefore be along this direction of travel. The procedure is thus as follows:

1. Determine the direction of travel: compute $-\dfrac{f'(x_0)}{f''(x_0)}$ to determine if it is left or right.
2. Travel along the direction until $f'(x) = 0$.
3. Assert this is the minima (or that the initial guess was bad).

In intuitive terms, the equation for Δx is saying the following: the shortest way to the nearest extrema can be determined by looking at concavity and slope. If the function is concave down, then by traveling in the direction of the derivative (positive being right, negative being left) you will reach the "top". Likewise, if the function is concave up, by traveling in the opposite direction than the derivative indicates you will reach the "bottom". Thus, provided our initial guess is within a concave up region, we will always track to the minima. This approach in one dimension has mostly the same pitfalls as the gradient technique: it could be that our initial guess was in a concave down region, in which case we will travel to the maxima, instead of the minima. The "local trap" phenomenon for local minima still happens, as well.

3.3 Two Dimensional Case

In two dimensions we must make use of the higher dimensional version of Taylor approximations. Given a function $f\colon \mathbb{R}^2 \to \mathbb{R}$, the two dimensional second order Taylor approximation about $\vec{x}_0$ is given by

$$f(\vec{x}_0 + \vec{s}) = f(\vec{x}_0) + \nabla f(\vec{x}_0)^T \vec{s} + \frac{1}{2!}\vec{s}^T H(\vec{x}_0)\vec{s}$$

where $H(\vec{x}_0)$ is the Hessian matrix evaluated at $\vec{x}_0$. The Hessian for $f(x, y)\colon \mathbb{R}^2 \to \mathbb{R}$ is defined as the 2×2 matrix $[\partial_{xy} f]$. Our problem is to determine the direction of $\vec{s}$. Assume that $f(\vec{x}_0 + \vec{s}) = f(\vec{x}_0)$; thus we have that

$$0 = \nabla f(\vec{x}_0)^T \vec{s} + \frac{1}{2!}\vec{s}^T H(\vec{x}_0)\vec{s}$$

$$\vec{s}^T = -2\nabla f(\vec{x}_0)^T H(\vec{x}_0)^{-1}$$

$$\vec{s} = (-2\nabla f(\vec{x}_0)^T H(\vec{x}_0)^{-1})^T = -2H(\vec{x}_0)^{-T}\nabla f(\vec{x}_0)$$

If we restrict our analysis to only functions that have continuous second partial derivatives (and moreover, to functions with nonsingular Hessian matrices), observe that the Hessian matrix becomes a symmetric matrix and so our equation becomes

$$\vec{s} = -2H(\vec{x}_0)^{-1}\nabla f(\vec{x}_0)$$

Thus given a starting point $\vec{x}_0$, we can compute the direction along which to travel. Just as with gradient descent, parametrize the line $\gamma(t) = \vec{x}_0 + t\vec{s}$ and perform root-finding on $f \circ \gamma(t)$.

The algorithm is a modification of the line search we used in the gradient descent method.

1. Determine the direction of travel from $\vec{x}_0$: $\vec{s} = -2H(\vec{x}_0)^{-1}\nabla f(\vec{x}_0)$
2. Travel from $\vec{x}_0$ using $\gamma\colon \mathbb{R} \to \mathbb{R}^2\colon \gamma(t) = \vec{x}_0 + t\vec{s}$
3. Compute t_1 such that $f'(\gamma(t_1)) = 0$, assign $\vec{x}_1 = \gamma(t_1)$ and check conditions for exit and reiterate if no exit condition is reached.

The exit conditions are the same as with the gradient method, as well:

1. If we exceed the in-built iteration limit, exit (computation time limit).
2. If our change between guesses is below the nearness threshold, exit (desired precision limit).
3. If we have that $f(\vec{x}_1) > f(\vec{x}_0)$, exit (unlucky guess).

3.4 Multidimensional Case

The Hessian and gradient are both defined for functions $f\colon \mathbb{R}^n \to \mathbb{R}$. The Hessian matrix of second partial derivatives will be an $n \times n$ matrix, but if we maintain the assumption of continuous second partials then it will still be symmetric. As a result, our line search will retain all of its properties in higher dimensional minima finding.

4 Conclusion

4.1 Method Comparison

Some of the disadvantages of Newton's method when compared with using only the gradient are as follows:

1. Additional assumption are required about differentiability; specifically, that the second partial derivatives are all continuous.
2. Computing the Hessian $H(\vec{x}_0)$ of f is required, which may be costly.
3. The matrix $H(\vec{x}_0)$ needs to be inverted, a costly procedure.
4. The matrix $H(\vec{x}_0)$ may be poorly conditioned, leading to numerical instability.

The advantages of Newton's method are in much faster convergence. While it is true that each iteration of Newton's method requires much more computation, in general it requires fewer iterations overall to converge to the desired result. As an example, figures 1b and 3a are the results of the gradient descent and Newton's method respectively for the same underlying function and starting guess. The gradient descent method exhibits a "zig-zagging" pattern that is typical of gradient descent line search, while Newton's method within two apparent iterations has already found its way to the desired result. Briefly, this is because the usage of the second order Taylor approximation is equivalent to approximating surfaces with the surface of a quadratic form and using the minima of that quadratic form as an estimate of the minima for the original function. Observe that the function $f(x, y) = xe^{-(x^2+y^2)}$ near the minima behaves much like a quadratic surface.

5 Appendix

5.1 Limitations

Both techniques tend to perform poorly near "flat" areas of the function; in regions where the function does not change in value much. In both cases, this is because local topography conveys rather limited information (can you tell which direction Mt. Everest is from where you are sitting?); for Newton's method, it is because quadratic surfaces are only good approximations of the function in regions near their maxima or minima. A useful example of this issue is the function $f(x, y) = \cos^2(x)e^y + 1$. The minima of this function occur for any y along lines of $x = \dfrac{(2k+1)\pi}{2}$; however, both techniques perform rather poorly for starting guesses near the origin.

5.2 Algorithm Sketch

Line search with gradient descent/Newton's method starting at $\vec{x}_n$:

0. If $f\colon \mathbb{R}^n \to \mathbb{R}^m$, use $F(\vec{x}) = f(\vec{x}) \cdot f(\vec{x})$ instead.
1. Compute the direction:
   (a) $\vec{s}_n = -\nabla f(\vec{x}_n)$ for gradient descent.
   (b) $\vec{s}_n = -2H(\vec{x}_n)^{-1}\nabla f(\vec{x}_n)$ for Newton's method.
2. Define the parametrized curve $\gamma_n(t) = \vec{x}_n + t\vec{s}_n$
3. Define the composition $h(t) = f \circ \gamma(t)$
4. Solve $h'(t) = 0$; t_n (so $f \circ \gamma(t_n)$ is an extrema).
5. Set $\vec{x}_{n+1} = \vec{x}_n + t_n\vec{s}_n$
6. Consult exit conditions:
   (a) Iteration count exceeded (out of time)
   (b) $f(\vec{x}_{n+1}) > f(\vec{x}_n)$ (wrong direction)
   (c) $|f(\vec{x}_{n+1}) - f(\vec{x}_n)|$ sufficiently small (nearness)
7. Reiterate from step 1.