Zero Order Minimization Of Quadratics (& Strictly Convex Functions)

Dhruv Malik

April 10th 2018

Overview. Consider the following equation for a general quadratic function:

$$f(x) = ax^2 + bx + c \tag{1}$$

Let's say we wanted to minimize this function, but we don't know the constants a, b or c. Hence, we cannot take a derivative. But let's say we were given zero order information. So, the only thing we have is oracle access to the function evaluated at points of our choice. I'm going to describe two algorithms with different convergence rates which ensure that we get within ϵ of the minimizer. I believe that both these algorithms will work for any strictly convex function. Indeed, I have an intuition that this might even work as a zero order method for our problem? Not sure about this. At the end I will have a short section explaining why I'm not sure how to actually solve the problem we wanted to solve, and where we might want to briefly look.

Method 1

Idea. The core idea is to descend down one side of the hill, without ever hopping across the minimizer except perhaps on the last iteration. Since we don't know how far we are from the minimum at any iteration, our step sizes need to be extremely small, to ensure that we don't lose track of which side of the minimizer we are currently on.

Algorithm. We want convergence to within ϵ of the minimizer. Determine whether you are on the left or right side of the minimizer by taking a step of ϵ to the left and right and seeing which way the cost decreases. Now, assume without loss of generality that our initial point x_1 is on the left side of x^*. Follow the update rule:

$$x_{t+1} = x_t + \epsilon \tag{2}$$

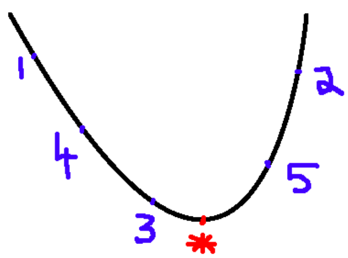
Follow this rule until we reach a particular iteration p such that $f(x_p) > f(x_{p-1})$. Then we have hopped to the right side of x^*, and can terminate the algorithm, by returning x_{p-1}. Since we know that until x_p the function values were monotonically decreasing, and we were only taking steps of size ϵ, we have that x_{p-1} is within ϵ of x^*.

Theorem 1. *If the distance from our initial point x_1 to the minimizer x^* is B, then we will converge to within ϵ of x^* in $\mathcal{O}(\frac{B}{\epsilon})$ iterations.*

Method 2

Idea. The core idea is that we want to always keep track of two points, which bracket the minimizer x^*. We then want to use a binary search like idea to find more points inside this interval which will still bracket x^*, but are closer to each other. This will keep reducing the size of the interval, until we are within ϵ of x^*.

Algorithm. Assume that we begin with two points x_1 and x_2 such that $x^* \in [x_1, x_2]$. Note that this is not hard to accomplish, one only needs to exponentially search ($z_{t+1} = 2z_t$) in two directions from an arbitrary starting point z_1. Then, pick x_3 to be the midpoint of x_1 and x_2, i.e. define $x_3 = \frac{x_1+x_2}{2}$. Similarly, define $x_4 = \frac{x_1+x_3}{2}$ and $x_5 = \frac{x_3+x_2}{2}$. An example of this is depicted in the picture below.



Two of the points $x_1 \ldots x_5$ will form our new interval. Consider the following cases:

1. If $f(x_4) > f(x_1)$, then define $a = x_1$ and $b = x_4$.
2. If $f(x_4) < f(x_1)$ and $f(x_3) > f(x_4)$, then $a = x_1$ and $b = x_3$.
3. If $f(x_4) < f(x_1)$ and $f(x_3) < f(x_4)$ and $f(x_3) < f(x_5)$, then $a = x_1$ and $b = x_5$.
4. If $f(x_4) < f(x_1)$ and $f(x_3) < f(x_4)$ and $f(x_3) > f(x_5)$, then $a = x_4$ and $b = x_2$.

Then reset $x_1 = a$ and $x_2 = b$ and repeat the algorithm to get a smaller interval.

Lemma 1. *At each iteration of this algorithm, the minimizer x^* always remains bracketed within the interval we pick.*

Proof. The proof follows from the fact that in the above algorithm, the cases are exhaustive and that $x^* \in [a, b]$. □

Lemma 2. *At each iteration of this algorithm, we reduce the size of our interval by a factor of at least $\frac{1}{4}$.*

Proof. The proof follows from the fact that in the above algorithm, we divide our big interval $[x_1, x_2]$ into four equally sized intervals, and then discard at least one of the four to get our new interval. □

Theorem 2. *If the size of our initial interval is B, then we converge to the minimizer x^* in $\mathcal{O}(\log(\frac{B}{\epsilon}))$ iterations.*

Proof. The proof follows from Lemma 1 (we always bracket the minimizer) and Lemma 2 (we are getting rid of at least a fourth of our interval every time). □

The Main Problem

The reason we went in this direction is because we want to minimize the RHS of following equation with respect to α_t:

$$\begin{aligned} f(x_{t+1}) &= f(x_t - \alpha_t w) && (3) \\ &\leq f(x_t) - \alpha_t w^\top \nabla f(x_t) + \frac{L}{2}\alpha_t^2 && (4) \end{aligned}$$

where w is some direction we randomly selected and we do not know the value of $\nabla f(x_t)$. But the issue is that we are only given evaluations of the LHS, and not the RHS. So, my methods don't directly translate to this problem because I don't have the right function evaluations. As such, I'm not really sure what to do.

That being said, I think my methods work for any bowl shaped/convex function, and since we are restricting our function to a line, my methods might give us something? Perhaps this is of independent interest to us for our problem? If we can prove that my methods will work on any function satisfying the PL Inequality (restricted to a line), then maybe we can use my zero order methods to minimize the LQR function along d dimensions, and then get a global minimizer?