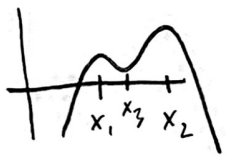
Honors Numerical Analysis

Lecture 7

Optimization

Recall: "root" finding solves $f(x) = 0$

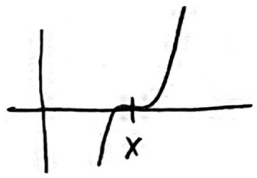
If we instead wanted to maximize/minimize f, we would look for x such that $f'(x) = 0$:



$f'(x_1) = 0$

$f'(x_2) = 0$

$f'(x_3) = 0$



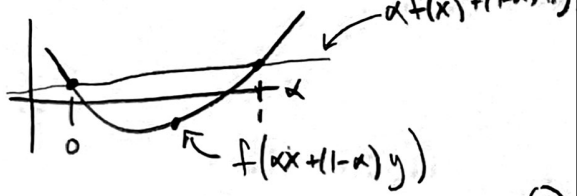
$f'(x) = 0$, but not a min/max

So we can apply our root finding methods to the equation $f'(x) = 0$ instead. Newton's Method becomes:

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

One (common) regime where we can guarantee a min/max is when f is convex:

Def f is convex on $[a,b]$ if $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$ for all $\alpha \in [0,1]$ and $x, y \in [a,b]$:



Thm If f is convex on $[a,b]$, then any local minimum of f is also a global minimum.

Proof (By contradiction) Let x^* be a local min of f, and $\tilde{x} \neq x^*$ be such that $f(\tilde{x}) < f(x^*)$.

Then, since f is convex, we have that

$$f\left(\alpha x^* + (1-\alpha)\tilde{x}\right) \leq \alpha f(x^*) + (1-\alpha) f(\tilde{x})$$

$$< \alpha f(x^*) + (1-\alpha) f(x^*)$$

$$= f(x^*)$$

But since f is convex, and x^* was assumed to be a local minimum, there exists some neighborhood around x^* such that $f(x) \geq f(x^*)$ for $x \in X^*$. But we just showed that $f(x) < f(x^*)$, and therefore we have a contradiction $\Rightarrow$ x^* is a global min.

Thm If f is strictly convex, then there is only one local minimum.

Ex:

convex

strictly convex

Multivariate Optimization

Now consider that we want to find

$$\vec{x}^* = \underset{\vec{x} \in \mathcal{X}}{\arg\min}\ f(x_1, x_2, \ldots, x_n)$$

Assume that f is smooth and strictly convex. Then we can (hopefully) solve this using Newton's Method on the system:

$$\left.\begin{aligned} \frac{\partial f}{\partial x_1} &= 0 \\ \frac{\partial f}{\partial x_2} &= 0 \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= 0 \end{aligned}\right\} \quad \nabla f = \vec{0}.$$

Let $\frac{\partial f}{\partial x_j} = f_j$. Then Newton's Method says:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \overline{J}^{-1}(\vec{x}^{(k)})\, \vec{f}(\vec{x}^{(k)})$$

where $\overline{J}_{jk} = \frac{\partial f_j}{\partial x_k}$ and $\vec{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$.

But since $f_j = \frac{\partial f}{\partial x_j}$ we have that

$$\overline{J}_{jk} = \frac{\partial}{\partial x_k}\left(\frac{\partial f}{\partial x_j}\right) = \frac{\partial^2 f}{\partial x_j \partial x_k} = H_{ij} \quad \leftarrow \text{the Hessian.}$$

So Newton is:

$$\boxed{\vec{x}^{(k+1)} = \vec{x}^{(k)} - H^{-1}(\vec{x}^{(k)})\, \nabla f(\vec{x}^{(k)})}$$

There is one main computational challenge with this: evaluating and inverting H is expensive.

Evaluation: $\theta\left(\frac{n^2}{2}\right)$

Inversion: $\theta(n^3)$ (We will come back to this "big oh" notation later.)

To avoid explicit construction of H, we consider <u>Quasi-Newton Methods</u> (Recall: Secant method was the most basic quasi-Newton method.)

<u>Broyden's Update</u>

The goal is to form an <u>approximation</u> to H on each step of Newton, and to be able to cheaply update this approximation:

<u>Example</u> $\vec{x}^{(1)} = \vec{x}^{(0)} - H^{(0)^{-1}} \nabla f(\vec{x}^{(0)})$

$\vec{x}^{(2)} = \vec{x}^{(1)} - H^{(1)^{-1}} \nabla f(\vec{x}^{(1)})$, $H^{(1)} = H^{(0)} + \underline{\text{something}}$

Recall Newton for the optimization problem, consider the implied linear approximation:

$$(*) \quad \nabla f(\vec{x}^{(k+1)}) \approx \vec{0} \approx \underbrace{\nabla f(\vec{x}^{(k)})}_{\nabla f^{(k)}} + \underbrace{H(\vec{x}^{(k)})}_{H^{(k)}} \underbrace{(\vec{x}^{(k+1)} - \vec{x}^{(k)})}_{\vec{s}^{(k)} = \text{Newton step}}$$

$$\Rightarrow H^{(k)} \vec{s}^{(k)} = -\nabla f^{(k)}$$

By (*), it should also be the case that (similar to secant method)

$$\nabla f^{(k+1)} - \nabla f^{(k)} \approx H^{(k+1)} \left(\vec{x}^{(k+1)} - \vec{x}^{(k)} \right)$$

$\Rightarrow$ $H^{(k+1)}$ should satisfy $H^{(k+1)} \vec{s}^{(k)} = \nabla f^{(k+1)} - \nabla f^{(k)}$

Of course this does not uniquely determine $H^{(k+1)}$ if $s^{(k)}$, $\nabla f^{(k+1)}$, and $\nabla f^{(k)}$ are known. Extra constraint:

$H^{(k+1)} - H^{(k)}$ be of rank one.

This gives:

$$H^{(k+1)} = H^{(k)} + \underbrace{\frac{1}{\vec{s}^{(k)T} \vec{s}^{(k)}}}_{\text{number}} \underbrace{\left(\nabla f^{(k+1)} - \nabla f^{(k)} - H^{(k)} \vec{s}^{(k)} \right) \vec{s}^{(k)T}}_{\text{outer product of two vectors, rank one.}}$$

Algorithm
- Initialize $H^{(0)}$ (finite differences, diagonal, etc.)
- set $\vec{x}^{(1)} = \vec{x}^{(1)} - H^{(0)^{-1}} \nabla f(\vec{x}^{(0)})$
- $H^{(1)} = H^{(0)} + \frac{1}{mn} (mn) m$
- $\vec{x}^{(2)} = \vec{x}^{(1)} - H^{(1)^{-1}} \nabla f(\vec{x}^{(1)})$
- etc.

Since we are really only interested in H^{-1}, is there a better way to compute $H^{(k+1)^{-1}}$ if we know $H^{(k)^{-1}}$?

Yes.

The updated H is of the form

$$H_1 = H_0 + \underbrace{\vec{u}\vec{v}^T}$$

rank-one update

Sherman-Morrison (-Woodbury) Formula

$$\left(A + \vec{u}\vec{v}^T\right)^{-1} = A^{-1} - \frac{A^{-1}\vec{u}\vec{v}^T A^{-1}}{1 + \vec{v}^T A^{-1}\vec{u}}$$

Proof (See Wikipedia)