6.006 Lecture 21 12/1/16

Topic Continuous Optimization II

Today

Applications of gradient descent

- Linear system solving

Newton's method (another iterative method for continuous optimization)

Applications:

- Computing roots
- Unconstrained minimization

Review of last time

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix} \in \mathbb{R}^n$$

Gradient Descent Method

step size can vary with k

start with x^0

Repeat

$$x^{k+1} = x^k - \eta_k \nabla f(x^k)$$

based on Taylor Expansion

$\approx \eta_k$

$$f(x) = \underbrace{f(x^i) + \nabla f(x^i)(x - x^i)}_{\text{linear approximation of } f} + \underbrace{\tfrac{1}{2}(x - x^i)^T \nabla^2 f(x^i)(x - x^i) + \dots}_{\text{"error" hopefully small}}$$

Known fact gradient descent converges to $\tilde{x}$

s.t. $\nabla f(\tilde{x}) = 0$

might not even be local min:
could be local max or saddle pt.
but all local mins are critical!
(if convex, all critical pts are local mins)

Linear Systems Solving

Given system of linear equations

$$
\begin{aligned}
a_{11} \cdot x_1 + a_{12} \cdot x_2 + \ldots + a_{1m} x_n &= b_1 \\
a_{21} \cdot x_1 + a_{22} \cdot x_2 + \ldots + a_{2n} \cdot x_n &= b_2 \\
&\vdots \\
a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \ldots + a_{mn} \cdot x_n &= b_m
\end{aligned}
$$

Find $x_1 \ldots x_n$ satisfying equations

OR equivalently:

Given $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$ $\quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \in \mathbb{R}^m$

Find $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n$

such that $Ax = b$

Today $m = n$ (A square matrix)
A invertible (soln exists)

How to solve?

1^{st} idea:

invert A:
compute A^{-1} & set $x^* = A^{-1}b$ ⇙ natural approach!

why not?

- can be slow (fastest algorithm uses $\Theta(n^{2.373})$ time)
- computing A^{-1} can involve very large numbers since might need to divide by small numbers
- if A is sparse (few non zero entries) then A^{-1} might not be

but can we hope to improve?

2^{nd} idea:

- phrase as unconstrained minimization problem
- use gradient descent!

Consider

$$f(x) = \frac{1}{2} x^T A x - bx \quad \Leftarrow \text{Quadratic function}$$

Why did we pick this crazy choice of f?
(what does it have to do with our goal?)

What is ∇f?

this is looking more promising!! $\Rightarrow$ $\nabla f(x) = Ax - b$

Note: $\nabla bx = b$
$\nabla (x^T B x) = 2Bx$

What does gradient descent do?

it finds an extremum – where $\nabla f(\tilde{x}) \approx 0$

if we find such an $\tilde{x}$ we have

$$0 \approx \nabla f(\tilde{x}) = A\tilde{x} - b$$

$$\text{so} \quad \underbrace{A\tilde{x} \approx b}_{\text{a solution!!}}$$

Comment: in fact, we picked f to be integral of equation we wanted to solve

Advantages:

- fast
- no division

Root finding

given $f: \mathbb{R}^n \to \mathbb{R}$

find x^* s.t. $f(x^*) = 0$ $\Leftarrow$ very different goal, right?

actually, we'll see that it is very related

e.g. $f(x) = x^2 - 2 \Rightarrow x^* = \pm\sqrt{2}$

idea 1 use gradient descent again

define $g(x) = \frac{x^3}{3} - 2x$ $\Leftarrow$ integral of f

$$g'(x) = \frac{3x^2}{3} - 2 = x^2 - 2$$

setting $g'(x) = x^2 - 2 = 0 \Rightarrow$ gradient descent solution $\tilde{x} = \pm\sqrt{2}$

idea 2 Newton's (also Newton-Raphson) method (also iterative approach!)

As before approx $f(x)$ using Taylor series:

$$f(x) = \underbrace{f(x^k) + f'(x^k)(x - x^k)}_{\hat{f}_k(x) \text{ is linear approx of } f \text{ around } x^k} + \underbrace{\tfrac{1}{2} f''(x^k)(x - x^k)^2 + \ldots}_{\text{error}}$$

since want x s.t. $f(x) = 0$:

$$0 = \hat{f}_k(x^{k+1}) = f(x^k) + f'(x^k)(x^{k+1} - x^k) \Leftarrow \text{set } \hat{f}_k(x^{k+1}) \text{ to } 0 \text{ \& solve for } x^{k+1}$$

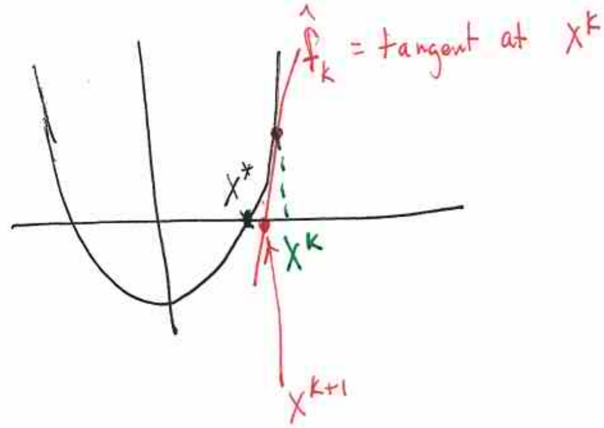
$$\Rightarrow x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$$

Newton's method for $n=1$ (Newton-Raphson)

start with x^0

repeat

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



General case $n \geq 1$

$$f(x) = \underbrace{f(x^k) + \nabla f(x^k)^T (x - x^k)}_{\hat{f}_k(x)} + \underbrace{\tfrac{1}{2}(x-x^k)^T \nabla^2 f(x^k)(x-x^k) + \ldots}_{\text{error}}$$

Setting

$$0 = \hat{f}_k(x) = f(x^k) + \nabla f(x^k)^T (x - x^k)$$

get $$x^{k+1} = x^k - \underbrace{\frac{f(x^k)}{\|\nabla f(x^k)\|^2} \nabla f(x^k)}_{= \frac{f(x^k)}{f'(x^k)} \text{ when } n=1}$$

note

f is non-linear & we are ignoring that

step size might not be small enough to make error small

$\Rightarrow$ convergence not as robust as for GD

i.e. might not converge

but if it does converge, then it is FAST

Turns out that error is being squared!

so if error >1, big problem

but if error <1, converges fast

Newton's method for square roots:

$$f(x) = x^2 - a$$

$$x^{(k+1)} = x^{(k)} - \left(\frac{\overbrace{(x^{(k)})^2 - a}^{f(x^{(k)})}}{\underbrace{2x^{(k)}}_{f'(x^{(k)})}} \right) = x^{(k)} - \frac{x^{(k)}}{2} + \frac{a}{2x^{(k)}} = \frac{1}{2}\Big[\underbrace{x^{(k)} + \frac{a}{x^{(k)}}}_{\text{average of } x^{(k)} + \frac{a}{x^{(k)}}}\Big]$$

dates back to Babylonians

Example $a = 2$

$x^{(0)} = 1.00000\ldots$

$x^{(1)} = 1.50000\ldots \qquad (1 + \frac{2}{1})/2 = 3/2$

$x^{(2)} = 1.41666\ldots \qquad (\frac{3}{2} + \frac{2}{3/2})/2 = 17/12$

$x^{(3)} = 1.414215686 \qquad 577/408$

$x^{(4)} = 1.414\ 213\ 562$

\# correct digits nearly doubles !!
(quadratic convergence)

Error analysis

Assume multiplicative error at stage k is $(1+\varepsilon_k)$

i.e. $x^{(k)} = \sqrt{a}\,(1+\varepsilon_k)$

then
$$x^{(k+1)} = \frac{x^{(k)} + \left(\frac{a}{x^{(k)}}\right)}{2}$$

$$= \frac{\sqrt{a}\,(1+\varepsilon_k) + \left(\frac{a}{\sqrt{a}\,(1+\varepsilon_k)}\right)}{2}$$

$$= \sqrt{a}\left[\frac{1+\varepsilon_k + \frac{1}{(1+\varepsilon_k)}}{2}\right] = \sqrt{a}\left[\frac{1+2\varepsilon_k+\varepsilon_k^2+1}{2(1+\varepsilon_k)}\right]$$

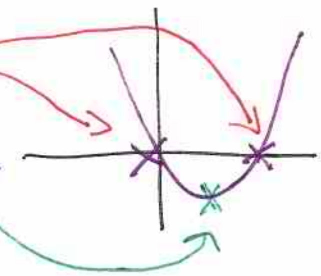
$$= \sqrt{a}\left[\frac{2+2\varepsilon_k+\varepsilon_k^2}{2(1+\varepsilon_k)}\right]$$

$$= \sqrt{a}\left[1 + \underbrace{\frac{\varepsilon_k^2}{2(1+\varepsilon_k)}}_{=\varepsilon_{k+1}}\right] \Leftarrow \text{quadratic convergence for } \varepsilon_0 < 1$$

Newton's method + unconstrained minimization are different!

one finds zeros,

the other finds minima



<u>But</u> Newton's method + unconstrained minimization are also very related!

- we already saw that minimization of "integral" can be used to find zeroes

- finding zeroes can be used to solve minimization too!

e.g. to find $x^* = \underset{x}{\operatorname{argmin}}\, f(x)$ for convex f

compute root of $g(x) = \|\nabla f(x)\|^2$

$$g(x^*) = 0 \Rightarrow \nabla f(x^*) = 0 \underset{\substack{\Uparrow \\ \text{convexity}}}{\Rightarrow} x^* = \underset{x}{\operatorname{argmin}}\, f(x)$$