Exam 3 next week

HW12

4 ch 1 → (hopefully) today

Goals.

1D opt methods

nD opt methods

~ nonlinear lsq.

Newton's Method

Reuse the Taylor approximation idea, but for optimization.

$x = x_k \rightsquigarrow \quad f(x+h) \approx f(x) + f'(x)h + f''(x)\frac{h^2}{2} =: \hat{f}(h)$

$0 \stackrel{!}{=} \hat{f}'(h) = f'(x) + f''(x)h \quad \rightsquigarrow h = -\frac{f'(x)}{f''(x)}$

$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$

Newton for solving

$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$

$\rightarrow$ locally quadr. conv. because equiv. to solving Newton

Demo: Newton's Method in 1D [cleared]

Steepest Descent/Gradient Descent

Given a scalar function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $\boldsymbol{x}$, which way is down?

Direction of steepest desc. $-\nabla f$

$\vec{x}_{k+1} = \vec{x}_k + \alpha s_k$ $\qquad s_k = -\nabla f(x_k)$

$\alpha \mapsto f(x_k + \alpha s_k)$ ← min. that "line search"

Empirically: linear conv.

Demo: Steepest Descent [cleared] (Part 1)

Steepest Descent: Convergence

Consider quadratic model problem:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$$

where A is SPD. (A good model of f near a minimum.)

$$e_{k+1} = \|x_{k+1} - x^*\| = \underbrace{\quad}\, e_k$$

Steepest Descent: Convergence

Consider quadratic model problem:

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{c}^T\boldsymbol{x}$$

where A is SPD. (A good model of f near a minimum.)

Define error $\boldsymbol{e}_k = \boldsymbol{x}_k - \boldsymbol{x}^*$. Then can show:

$$||\boldsymbol{e}_{k+1}||_A = \sqrt{\boldsymbol{e}_{k+1}^T A\boldsymbol{e}_{k+1}} = \frac{\sigma_{\max}(A) - \sigma_{\min}(A)}{\sigma_{\max}(A) + \sigma_{\min}(A)}||\boldsymbol{e}_k||_A$$

$\rightarrow$ confirms linear convergence.

Convergence constant related to conditioning:

$$\frac{\sigma_{\max}(A) - \sigma_{\min}(A)}{\sigma_{\max}(A) + \sigma_{\min}(A)} = \frac{\kappa(A) - 1}{\kappa(A) + 1}.$$

Hacking Steepest Descent for Better Convergence

Extrapolation methods:

Heavy ball method:

Demo: Steepest Descent [cleared] (Part 2)

Hacking Steepest Descent for Better Convergence

Extrapolation methods:

Look back a step, maintain '*momentum*'.

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k) + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

Heavy ball method:

constant $\alpha_k = \alpha$ and $\beta_k = \beta$. Gives:

$$||\boldsymbol{e}_{k+1}||_A = \frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} ||\boldsymbol{e}_k||_A$$

Demo: Steepest Descent [cleared] (Part 2)

Optimization in Machine Learning

What is *stochastic gradient descent* (*SGD*)?

Optimization in Machine Learning

What is *stochastic gradient descent* (*SGD*)?

Common in ML: Objective functions of the form

$$f(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}),$$

where each f_i comes from an *observation* ("data point") in a (training) data set. Then "*batch*" (i.e. normal) gradient descent is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha\frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{x}_k).$$

Stochastic GD uses one (or few, "*minibatch*") observation at a time:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha\nabla f_{\phi(k)}(\boldsymbol{x}_k).$$

ADAM optimizer: GD with exp. moving avgs. of ∇ and its square.

Conjugate Gradient Methods

Can we optimize in *the space spanned* by the last two step directions?

Demo: Conjugate Gradient Method [cleared]

Conjugate Gradient Methods

Can we optimize in *the space spanned* by the last two step directions?

$$(\alpha_k, \beta_k) = \text{argmin}_{\alpha_k,\beta_k} \left[f\left(\boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k) + \beta_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \right) \right]$$

- Will see in more detail later (for solving linear systems)
- Provably optimal first-order method for the quadratic model problem
- Turns out to be closely related to Lanczos (A-orthogonal search directions)

Demo: Conjugate Gradient Method [cleared]

orth: $x^T y = 0$ A-orth: $\vec{x}^T A \vec{y} = 0$

A spd

$$A = Q D Q^T$$

$\uparrow$

diag > 0

$$\underset{\sim}{x}^T A \underset{\sim}{y} = (Q^T x) D (Q^T y)$$

Nelder-Mead Method

Idea:

simplex gymnastics

Demo: Nelder-Mead Method [cleared]

Newton's method (n D)

$f: \mathbb{R}^n \to \mathbb{R}$ ∇

$\hookrightarrow$ Newton in 1D: solve $f' = 0$

What does Newton's method look like in n dimensions?

$\hookrightarrow$ in nD: use N. to solve $\nabla f = 0$

$$f(\vec{x}+\vec{s}) \approx \underbrace{f(\vec{x}) + \nabla f(\vec{x})^T \vec{s} + \frac{1}{2}\vec{s}^T H_f(\vec{x})\vec{s}}_{\hat{f}(\vec{s})}$$

$$0 = \nabla \hat{f}(\vec{s}) \rightsquigarrow \quad H_f(\vec{x})\vec{s} = -\nabla f(\vec{x})$$

$$\vec{x}_{k+1} = \vec{x}_k - H_f(\vec{x})^{-1} \nabla f(\vec{x})$$

$$\nabla \hat{f}(\vec{s}) = \nabla f(\vec{x})^T + \underbrace{\vec{s}^T H_f(\vec{x})}_{?} ? \qquad \frac{\partial \hat{f}}{\partial s_i} =$$

Newton's method (n D): Observations

Drawbacks?

- Need 2 derivatives
- expensive: need Hessian solve
- dependent on cond. of Hessian

Demo: Newton's Method in n dimensions [cleared]

Quasi-Newton Methods

Secant/Broyden-type ideas carry over to optimization. How?
Come up with a way to update to update the approximate Hessian.

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(\vec{x})$$

$\alpha_k \to$ line search parameter

$$\vec{s}_k = \vec{x}_{k+1} - \vec{x}_k$$

$$\vec{y}_k = \nabla f(\vec{x}_{k+1}) - \nabla f(\vec{x}_k)$$

$$B_{k+1} s_k = y_k$$

$\leftarrow$ secant condition

BFGS: Secant-type method, similar to Broyden:

$$B_{k+1} = B_k + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} - \frac{B_k \boldsymbol{s}_k \boldsymbol{s}_k^T B_k}{\boldsymbol{s}_k^T B_k \boldsymbol{s}_k}$$

Nonlinear Least Squares: Setup

What if the f to be minimized is actually a 2-norm?

$$f(\boldsymbol{x}) = \|\boldsymbol{r}(\boldsymbol{x})\|_2, \qquad \boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{y} - \boldsymbol{a}(\boldsymbol{x})$$