November 7, 2024

## Announcements

- HW 7
- HW 8

## Goals

## Review

# Unimodality

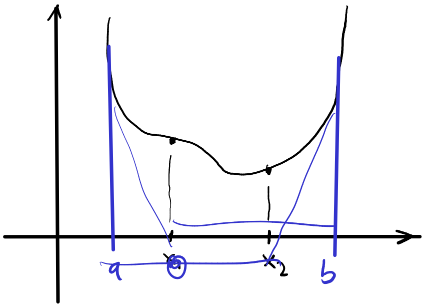
Would like a method like bisection, but for optimization.
In general: No invariant that can be preserved.
Need *extra assumption*.

$x^*$ in $[a,b]$ so that for $x_1 < x_2 \in [a,b]$

- $x_2 < x^* \Rightarrow f(x_1) > f(x_2)$
- $x^* < x_1 \Rightarrow f(x_1) < f(x_2)$

# Golden Section Search

Suppose we have an interval with $f$ unimodal:

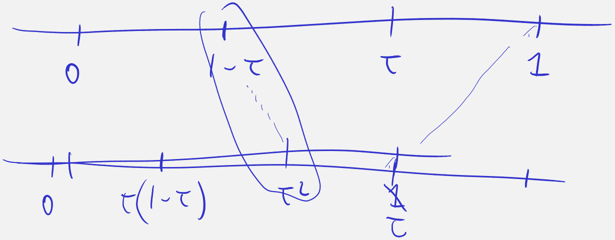



Would like to maintain unimodality.

- If $f(x_1) < f(x_2)$, reduce to $(a, x_2)$
- If $f(x_1) \geq f(x_2)$, reduce to $(x_1, b)$

# Golden Section Search: Efficiency

Where to put $x_1$, $x_2$?




$$\tau^2 = 1 - \tau \qquad \tau = (\sqrt{5} - 1)/2$$

Convergence rate?

Linear

# Newton's Method

Reuse the Taylor approximation idea, but for optimization.

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

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

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

**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?

$-\nabla f$ is the steepest descent dir.

- Use 1D opt. on the line search problem

$$\alpha \mapsto f(x_k + \alpha \nabla f(x_k))$$

- Find $\alpha_{min}$
- $x_{k+1} = x_k + \alpha_{min} \nabla f(x_k)$.

**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.)

## 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:

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

where $\|\boldsymbol{x}\|_A = \sqrt{\boldsymbol{x}^T A\boldsymbol{x}}$. $\to$ 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:

$$x_{k+1} = x_k + \alpha_k \nabla f(x_k) + \beta_k (x_k - x_{k-1})$$

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:

For specific constant $\alpha_k = \alpha$ and $\beta_k = \beta$, can attain:

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

**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 $\nabla$ and its square.

# Conjugate Gradient Methods

orth $\vec{x}_i^T \vec{x}_j = 0$ / conj. $\vec{x}_i^T A \vec{x}_j = 0$ SPD

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

**Demo:** Conjugate Gradient Method [cleared]

# Conjugate Gradient Methods

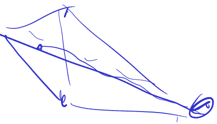
Can we optimize in *the space spanned* by the last two step directions?

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

- 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]

# Nelder-Mead Method



Idea:

Simplex gymnastics

**Demo:** Nelder-Mead Method [cleared]

# Newton's method ($n$ D)

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

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

$\nabla \hat{f}(\vec{s}) = 0$ to find

$$\nabla \hat{f}(\vec{s}) = \nabla f(\vec{x}) + H_f(\vec{x})\vec{s} = 0$$

$$\Rightarrow \quad s = -H_f(\vec{x})^{-1}\nabla f(\vec{x})$$

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

# Newton's method ($n$ D): Observations

Drawbacks?

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

# Newton's method ($n$ D): Observations

Drawbacks?

- Need second (!) derivatives
  (addressed by Conjugate Gradients, later in the class)
- local convergence
- Works poorly when $H_f$ is nearly indefinite

**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.

BFGS: Secant-type method, similar to Broyden:

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