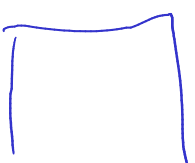
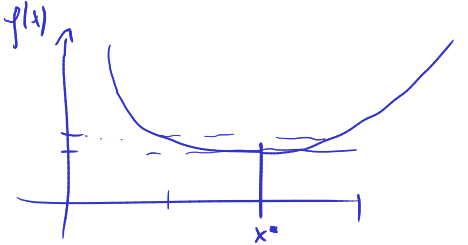
- Changes:
  - no AK office hours today ✓
  - very short AK office hours Tue ✓
  - no AK office hours Thu / Luke teaching
  - no class Nov 8 ✓
- 4-credit assignment 1 posted
- Exam 1 ongoing

$$Ax = u(v^T x)$$

$$= uv^T$$

$$= uv^T$$

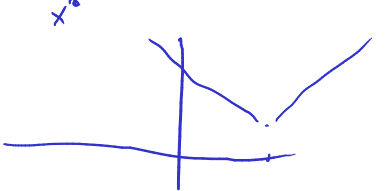
$A =$ 



$$f'(x^*) = 0$$

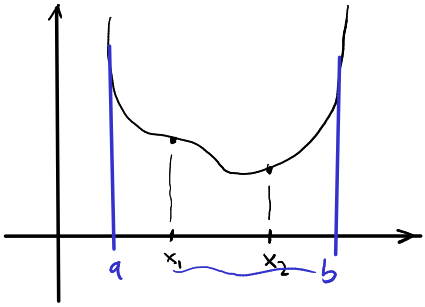
unimodal:

$$f(x^* + h) = f(x^*) + f''(x^*)\frac{h^2}{2}$$



Golden Section Search

Suppose we have an interval with f unimodal:



Would like to maintain unimodality.

Pick x_1, x_2

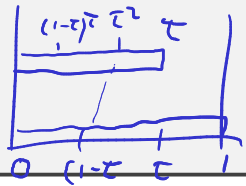
If $f(x_1) > f(x_2)$, reduce to (x_1, b)

If $f(x_1) \leq f(x_2)$, reduce to (a, x_2)

Golden Section Search: Efficiency

Where to put x_1, x_2?

$$x_1 = a + (1-\tau)(b-a)$$
$$x_2 = a + \tau(b-a)$$



Demo: Golden Section Proportions [cleared]

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

"golden section search"

Convergence rate?

linear

Newton's Method

Solve $f'(x) = 0$

Reuse the Taylor approximation idea, but for optimization.

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

$$\hat{f}'(h) = f'(x) + f''(x)\cdot h \doteq 0$$

$$h = -\frac{f'(x)}{f''(x)}$$

$x_0 =$ (starting guess)

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

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?

Direction of steepest descent: $-\nabla f$

- $\vec{x}_0 =$ (starting guess)
- $\vec{s}_n = -\nabla f(\vec{x}_n)$
- Use a line search to choose α_n to minimize the function $\alpha \mapsto f(\vec{x}_n + \alpha \vec{s}_n)$
- $\vec{x}_{n+1} = \vec{x}_n + \alpha_n \vec{s}_n$

Demo: Steepest Descent [cleared] (Part 1)

Steepest Descent: Convergence

$x^T A x$

Consider quadratic model problem:

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

$\begin{pmatrix} 0.5 & 0 \\ 0 & 2.5 \end{pmatrix}$

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

Define error $\vec{e}_k = \vec{x}_k - \vec{x}^*$

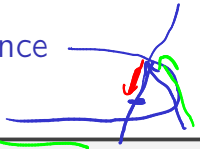
$\|x\|_A = x^T A x$ (only spd)

$$\|\vec{e}_{k+1}\|_A = \sqrt{\vec{e}_{k+1}^T A \vec{e}_{k+1}}$$

$$= \frac{\sigma_{max}(A) - \sigma_{min}(A)}{\sigma_{max}(A) + \sigma_{min}(A)} \|\vec{e}_k\|_A$$

$$= \frac{\kappa(A) - 1}{\kappa(A) + 1} \cdot \|\vec{e}_k\|_A$$

Hacking Steepest Descent for Better Convergence



Extrapolation methods:

$$x_{n+1} = x_n - \alpha_n \nabla f(x_n) + \beta_n (x_n - x_{n-1})$$

SD

Heavy ball method:

$$\alpha_n = \alpha \qquad \beta_n = \beta$$

Demo: Steepest Descent [cleared] (Part 2)

Optimization in Machine Learning

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

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

"batch" $\quad \vec{x}_{n+1} = x_n - \alpha \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x_n)$

"mini batch"

same part of this sum

"ADAM" moving averages of ∇ and the square of the gradient

Conjugate Gradient Methods

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

$$(\alpha_k, \beta_k) = \underset{\alpha, \beta}{\operatorname{argmin}} \left[f\left(x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1})\right) \right]$$

Demo: Conjugate Gradient Method [cleared]

Nelder-Mead Method

Idea:

Demo: Nelder-Mead Method [cleared]