CSC338 Numerical Methods

Lecture 11

March 25, 2020

About these slides

These slides are meant to be presentation aid, *not* a source of information.

Please use these slides in conjunction with the notes and the slides that comes with the textbook.

Unconstrained Optimization

Find a local minimum of $f : \mathbb{R} \to \mathbb{R}$

Approach: Golden Section Search

If f is unimodal on [a, b] then we can iteratively shrink the interval in which the minima $x^\star$ lies in

Unconstrained Optimization

Find a local minimum of $f : \mathbb{R} \to \mathbb{R}$

Approach: Newton's Method

Approximate $f(x)$ using a quadratic function, and find a critical point of the approximation.

Today

Find a local minimum of $f : \mathbb{R}^n \to \mathbb{R}$

We'll talk about:

- Newton's Method
- Gradient Descent
- Reading Contour Plots

Newton's Method for $f : \mathbb{R}^n \to \mathbb{R}$

When $f : \mathbb{R} \to \mathbb{R}$, we have the Taylor Series Expansion:

The result extends to $f : \mathbb{R}^n \to \mathbb{R}$

Newton's Method Idea

In each iteration, we have an estimate $\mathbf{x}_k$ of a minimum of f.

So we approximate f(x) with

Newton's Method Update Rule

One major disadvantage of Newton's Method is that computing the Hessian $H_f(\mathbf{x})$ is very expensive! (Recall $H_f(\mathbf{x}) \in \mathbb{R}^{n \times n}$)

Newton's Method Example

We wish to find a local minimum of
$f(\mathbf{x}) = x_1^4 + x_1^2 x_2 + x_1^2 + 2x_2^2 + x_2$, starting with $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

First, compute $\nabla_{\mathbf{x}} f(\mathbf{x})$ and $H_f(\mathbf{x})$

Newton's Method Example

We wish to find a local minimum of $f(\mathbf{x}) = x_1^4 + x_1^2 x_2 + x_1^2 + 2x_2^2 + x_2$, starting with $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

First, compute $\nabla_{\mathbf{x}} f(\mathbf{x})$ and $H_f(\mathbf{x})$

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} 4x_1^3 + 2x_1 x_2 + 2x_1 \\ x_1^2 + 4x_2 + 1 \end{bmatrix}$$

$$H_f(\mathbf{x}) = \begin{bmatrix} 12x_1 + 2x_2 + 2 & 2x_1 \\ 2x_1 & 4 \end{bmatrix}$$

Newton's Method Example

Plug in $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. What are the values of $\nabla_{\mathbf{x}} f(\mathbf{x}_0)$ and $H_f(\mathbf{x}_0)$?

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} 4x_1^3 + 2x_1x_2 + 2x_1 \\ x_1^2 + 4x_2 + 1 \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \end{bmatrix}$$

$$H_f(\mathbf{x}) = \begin{bmatrix} 12x_1 + 2x_2 + 2 & 2x_1 \\ 2x_1 & 4 \end{bmatrix} = \begin{bmatrix} \qquad & \qquad \\ \qquad & \qquad \end{bmatrix}$$

Newton's Method Example

Plug in $\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. What are the values of $\nabla_{\mathbf{x}} f(\mathbf{x}_0)$ and $H_f(\mathbf{x}_0)$?

$$\nabla_{\mathbf{x}} f(\mathbf{x}_0) = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$
$$H_f(\mathbf{x}_0) = \begin{bmatrix} 16 & 2 \\ 2 & 4 \end{bmatrix}$$

Newton's Method Example

We need $\mathbf{s}_0$ so that $H_f(\mathbf{x}_0)\mathbf{s}_0 = -\nabla_{\mathbf{x}} f(\mathbf{x}_0)$.

Solve for $\mathbf{s}_0$ in:

$$\begin{bmatrix} 16 & 2 \\ 2 & 4 \end{bmatrix} \mathbf{s}_0 = - \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

Use Gauss Elimination!

Newton's Method Example

We need $\mathbf{s}_0$ so that $H_f(\mathbf{x}_0)\mathbf{s}_0 = -\nabla_{\mathbf{x}} f(\mathbf{x}_0)$.

Solve for $\mathbf{s}_0$ in:

$$\begin{bmatrix} 16 & 2 \\ 2 & 4 \end{bmatrix} \mathbf{s}_0 = - \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

Use Gauss Elimination!

We get

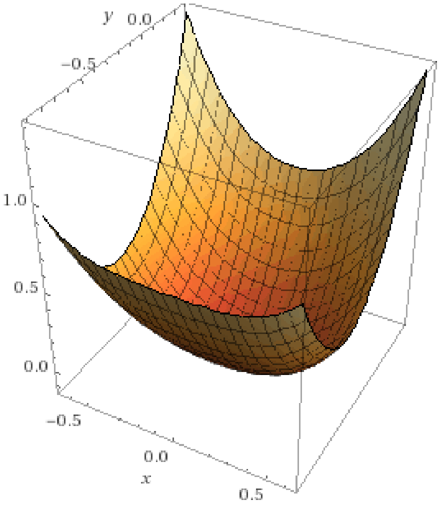
$$\mathbf{s}_0 = \begin{bmatrix} -\frac{2}{5} \\ -\frac{4}{5} \end{bmatrix}$$

Newton's Method Update

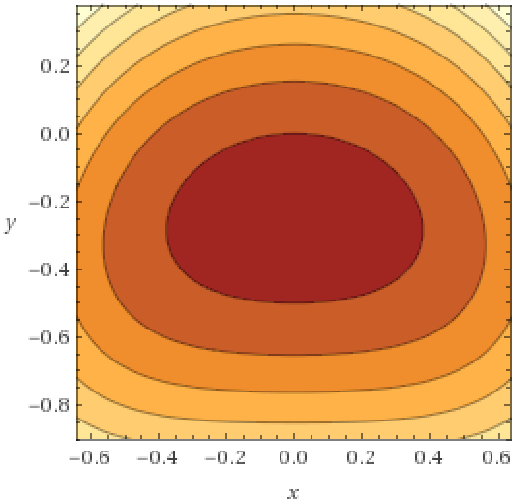
How do we compute $\mathbf{x}_1$ given

$$\mathbf{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \; \mathbf{s}_0 = \begin{bmatrix} -\frac{2}{5} \\ -\frac{4}{5} \end{bmatrix}?$$

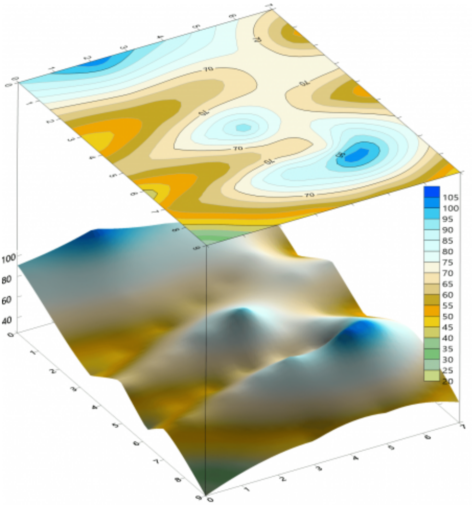
Function 3D Plot



Contour Plot



How to read contour plots



Steepest Descent

Steepest Descent / Gradient Descent

Key idea:

- The *gradient* of a differentiable function points *uphill*
- The *negative gradient* of a differentiable function points *downhill*

Example: $f : \mathbb{R} \to \mathbb{R}$

Take $f(x) = x^2$

Example: $f : \mathbb{R}^2 \to \mathbb{R}$

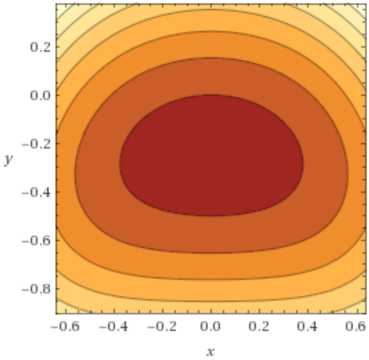
Take $f(\mathbf{x}) = x_1^4 + x_1^2 x_2 + x_1^2 + 2x_2^2 + x_2$,

At:

$$\mathbf{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \; \nabla_{\mathbf{x}} f(\mathbf{x}) =$$

$$\mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; \nabla_{\mathbf{x}} f(\mathbf{x}) =$$

Contour Plot



Note: The gradient is always perpendicular to the contour!

Why does this work?

Intuition, a function $f : \mathbb{R}^n \to \mathbb{R}$ locally looks like a plane.

In other words, locally we can approximate f using

It turns out that $-\nabla f(\mathbf{x})$ is, locally, the direction of the steepest descent.

Steepest descent

Algorithm to find a minima of $f : \mathbb{R}^n \to \mathbb{R}$ locally

Start from an initial guess $\mathbf{x}_0$ and update:

Steepest descent pros & cons

Steepest descent example

Show slide 29 and 30

Steepest Descent vs Newton's Method

Steepest Descent:

Newton:

Quasi-Newton Methods

Steepest Descent: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)$

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

Quasi-Newton:

Where B_k is an approximation of the Hessian matrix.

Homework Grade Prediction Revisited

Recall the problem of predicting a student's hw3 grade given their hw1 and hw2 grades.

$$A = \begin{bmatrix} a_1^{(1)} & a_2^{(1)} \\ a_1^{(2)} & a_2^{(2)} \\ \vdots & \vdots \\ a_1^{(73)} & a_2^{(73)} \end{bmatrix} \qquad b = \begin{bmatrix} b_1^{(1)} \\ b_1^{(2)} \\ \vdots \\ b_1^{(73)} \end{bmatrix}$$

Problem: Find $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ to minimize $||A\mathbf{x} - \mathbf{b}||_2$

We can treat this as a non-linear optimization problem!

Grade Prediction as Non-Linear Optimization

Define

$$
\begin{aligned}
f(\mathbf{x}) &= ||A\mathbf{x} - \mathbf{b}||_2 \\
&= (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) \\
&=
\end{aligned}
$$

Computing Gradient

Now, given that

$$f(\mathbf{x}) = \sum_{j=1}^{73} (a_1^{(j)} x_1 + a_2^{(j)} x_2 - b^{(j)})^2$$

Let's compute $\nabla_{\mathbf{x}} f(\mathbf{x})$:

Gradient Descent

Start with some $\mathbf{x}_0$, e.g. $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ or $\mathbf{x}_0 = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$. (Why?)

Then take gradient descent steps:

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

until $\mathbf{x}_{k+1}$ is sufficiently close to $\mathbf{x}_k$, or until $f(\mathbf{x}_{k+1})$ is sufficiently close to $f(\mathbf{x}_k)$

Why Gradient Descent?

Instead of computing $\nabla_{\mathbf{x}} f(\mathbf{x})$ exactly, we can estimate the gradient using a small subset of our data (subset of 73 students)

Gradient descent works for more complicated functions, like neural networks!