Steepest Descent

Wednesday, October 4, 2023 1:28 PM

AKA. Gradient Descent

Warm-up:

How many pts do you need to make a quadratic?

3pt

(x_0, y_0), (x_1, y_1), (x_2, y_2)

How do we construct the quadratic?

option 1: $y = a(x-\alpha)^2 + \beta$ $\quad (\alpha, \beta) = \text{vertex}$

option 2: we know $y = ax^2 + bx + c$ is a form for a quadratic.

Plug in pts & get a linear system

$$y_0 = a x_0^2 + b x_0 + c$$
$$y_1 = a x_1^2 + b x_1 + c$$
$$y_2 = a x_2^2 + b x_2 + c$$

solve this for $a, b, \& c$

option 3: $P_2(x) = y_0 \dfrac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \dfrac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}$

$$+ y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}$$

$$+ y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

Goal: Find the root of a nonlinear system of equations.

Why do we need this?

- Newton is very sensitive to the initial guess
- basin of convergence is difficult to find.

IDEA: Steepest descent

Our goal is to find $\bar{\alpha} = (\alpha_1, \cdots, \alpha_n)^T$ st

$$\begin{cases} f_1(\bar{\alpha}) = 0 \\ f_2(\bar{\alpha}) = 0 \\ \vdots \\ f_n(\bar{\alpha}) = 0 \end{cases}$$

We will recast this problem as finding the min. of a quadratic

$$g(x_1, \cdots x_n) = \sum_{j=1}^{n} (f_j(\bar{x}))^2$$

$$g(x_1, \cdots x_n) = \sum_{j=1}^{m} (f_j(\bar{x}))^2$$

where $\bar{x} = (x_1, \cdots, x_n)^T$

We KNOW $\bar{\alpha}$ is a min of g

WARNING: g will most likely have multiple minimums

So verify that your min. is a root of the system of equations. (Check!)

Newton: $x_{k+1} = x_k - (J(x_k))^{-1} F(x_k)$

x_k: previous; $-(J(x_k))^{-1} F(x_k)$: update

The technique for building our iteration is to start a pt on the surface & follow the path of maximum decrease. (Hence steepest descent)

What is the direction of max. decrease?

negative gradient; i.e. $-\nabla g$

Rough Outline

1- Start w/ initial guess x_0
2- Determine the direction to go that will result in a max decrease in g
3- Move an "appropriate" amount β in that direction & call the new location x_1
4- Continue at step 2 until convergent.

Step 3 is vague. There are many ways of picking β.

let's figure out how to evaluate ∇g interms of the f_j's

We know $g(x) = \sum_{j=1}^{n} (f_j(x))^2$

$$\nabla g = \left(\frac{dg}{dx_1}, \frac{dg}{dx_2}, \dots \frac{dg}{dx_n} \right)$$

Check box $x = (x_1, x_2)$

$g(x) = (f_1(x))^2 + (f_2(x))^2$

let's start w/

$$\frac{dg}{dx_i} = 2\left(f_1(x) \frac{df_1}{dx_i} + f_2(x) \frac{df_2}{dx_i} + \dots + f_n \frac{df_n}{dx_i} \right)$$

$$= 2 \begin{bmatrix} \frac{\partial f_1}{\partial x_i} & \frac{\partial f_2}{\partial x_i} & \cdots & \frac{\partial f_n}{\partial x_i} \end{bmatrix} \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

$\Rightarrow$ We can write ∇g as a matrix vector multiplication.

$$\nabla g = 2 \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_1} \\ \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_2} & & \\ \vdots & & & \\ \frac{\partial f}{\partial x_n} & & & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

$$= 2 J^T F$$

Our iteration

$$x_{k+1} = x_k - \lambda_k \nabla g(x_k)$$

$$= x_k - \underbrace{\lambda_k}_{\text{scalar multiple of } -\nabla g \text{ direction}} 2 J_k^T F(x_k)$$

Notation: $J_k = J(x_k)$

Our new (temp) goal is to find λ_k so that $g(x_k - \lambda_k 2 J_k^T F(x_k))$ is minimized. This is a scalar minimization problem.

minimized. This is a scalar minimization problem.

How far do we go in the $-\nabla g$ direction?
i.e. what is λ_k?

let define $h(\lambda_k) = g(x_k - \lambda_k \nabla g(x_k))$

Our goal is find λ_k so that this minimized.

Finding the exact min is hard since $h(\lambda)$ is a "complex" non-linear function.
So people approximate the minimum.

option 1: line search.

textbook { option 2: approx. $h(\lambda)$ with a quadratic. & find the min.

Project option is to explore different methods

let $p(\beta)$ denote my quadratic approx of $h(\lambda)$

where

When is $p(\beta)$ minimized? where $p'(\beta) = 0$

let's pick our 3 pts.

Choose $\lambda_1 = 0$ $\quad \lambda_1 \leq \lambda_2 \leq \lambda_3$

$$(\lambda_1, \underbrace{h(\lambda_1)}_{h_1}) = (0, g(x_k))$$

we know our quadratic can be written as

$$p(\beta) = h_1 + s_1 \beta + s_3 \beta (\beta - \lambda_2)$$

(Newton - Divided difference)

where $h_2 = h(\lambda_2)$ $\quad h_3 = h(\lambda_3)$

$$s_1 = \frac{h_2 - h_1}{\lambda_2} \quad s_2 = \frac{h_3 - h_2}{\lambda_3 - \lambda_2} \quad \& \quad s_3 = \frac{s_2 - s_1}{\lambda_3}$$

The min. of quad. $p(\beta)$ happens at

$$\beta = \frac{-s_1 + \lambda_2 s_2}{2 \lambda_3} \quad \left(\text{where } p'(\beta) = 0\right)$$

What are λ_2 & λ_3?

Take $\lambda_2 = \frac{1}{2}(\lambda_1 + \lambda_3)$ to the midpt

So we just need to pick λ_3.

IDEA: Set $\lambda_3 = 1$

Check if concave up
if not set $\lambda_3 = 0.5\lambda_3$

Repeat until you have a concave up. polynomial.

Pseudocode: Steepest Descent

Input: $\bar{F}(x)$ = function we want the root of
$\bar{x}_0$ = initial guess
N_{max} = max # of iterations
tol = tolerance
J = Jacobian of F.

Output: x^* = approximate root
ier = error message
its = # of iterations

Steps:

Step 1: if $\|F(x_0)\| == 0$

$x^* = x_0$

ier = 0

return

Step 2: $J_0 = J(x_0)$

$F_0 = F(x_0)$

Find β = step length

$x_1 = x_0 - 2\beta J_0^+ F_0$

it = 1 = counter

Step 3: While $|x_0 - x_1| > tol$

Set $x_0 = x_1$

$J_0 = J(x_0)$

$F_0 = F(x_0)$

Find β = step length

$x_1 = x_0 - 2\beta J_0^+ F_0$

it = it + 1

if it $\geq$ Nmax

$x^* = x_1$

ier = 1

return

Step 4: $x^* = x_1$

ier = 0

Step 4. $x = x_1$
ier = 0
return

Possible problems:

- $g(x)$ may have many minimums or you may get stuck

Since we have a root finding problem, we should check if $\|F(x^*)\| < \varepsilon$.

- It is possible that you go back to where you started or places you have been.

- If this converges, it is slow (nearly linear)