

Lecture 3 2023-04-02

Last time: propagating ODEs

Today: smooth, unconstrained optimization

$$\min_x f(x)$$

subject to: $g_i(x) \leq 0 \quad i = 1, \dots, n_{ineq}$
 $h_i(x) = 0 \quad i = 1, \dots, n_{eq}$

gradient: if $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and $f \in \mathcal{C}^1$
i.e., f has a 1st

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \in \mathbb{R}^n$$

derivative

Hessian: if $f: \mathbb{R}^n \rightarrow \mathbb{R}$ and $f \in \mathcal{C}^2$
 f has 2nd
derivative

$$H = \nabla_{xx} f = \begin{pmatrix} \partial^2 f / \partial x_1^2 & \partial^2 f / \partial x_1 \partial x_2 & \dots & \partial^2 f / \partial x_1 \partial x_n \\ \vdots & & & \\ \partial^2 f / \partial x_n \partial x_1 & \dots & \dots & \partial^2 f / \partial x_n^2 \end{pmatrix}$$

symmetric matrix in $\mathbb{R}^{n \times n}$

usage: construct 2nd-order approximation about \bar{x}

$$f(x) \approx f(\bar{x}) + \nabla_x f(\bar{x})^T \delta x + \frac{1}{2} \delta x^T \nabla_{xx} f(\bar{x}) \delta x$$

+ higher order terms

e.g. $f(x) = c^T x$ where $c \in \mathbb{R}^n$

$$\rightarrow \nabla_x f(x) = c \in \mathbb{R}^n$$

Jacobren: if $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $f \in \mathcal{C}^1$

$$J = \begin{pmatrix} \partial f_1 / \partial x & \partial f_1 / \partial x_1^2 & \partial f_1 / \partial x_1 \partial x_n \\ \vdots & \vdots & \vdots \\ \partial f_m / \partial x & \partial f_m / \partial x_m \partial x_1 & \partial f_m / \partial x_m \partial x_n \end{pmatrix}$$

usage: $f(x) \approx f(\bar{x}) + J \delta x + \text{H.O.T.}$

e.g. if $A = \begin{pmatrix} -a_1 - \\ \vdots \\ -a_m - \end{pmatrix} \in \mathbb{R}^{m \times n}$

$$f(x) = Ax \rightarrow \frac{\partial f}{\partial x} = A$$

$$\begin{aligned}
 & \min_x f(x) \\
 \mathcal{P} \quad & \text{subj. to: } g_i(x) \leq 0 \quad i = 1, \dots, n_{\text{ineq}} \\
 & h_i(x) = 0 \quad i = 1, \dots, n_{\text{eq}}
 \end{aligned}$$

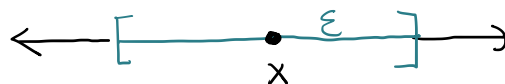
feasible set for \mathcal{P} :

$$\begin{aligned}
 \mathcal{L} = \{ x \in \mathbb{R}^n \mid & g_i(x) \leq 0 \quad \forall i \in [1, n_{\text{ineq}}] \\
 & \text{and } h_i(x) = 0 \quad \forall i \in [1, n_{\text{eq}}] \}
 \end{aligned}$$

ε -ball:

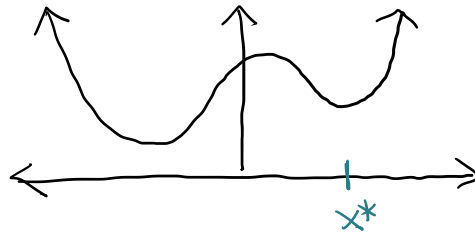
$$B_\varepsilon(x) = \{ y \in \mathbb{R}^n \mid d(x, y) \leq \varepsilon \}$$

scalar case



Local minimizer: $x^* \in \mathcal{L}$ is a local minimizer of

\mathcal{P} if $\exists B_\varepsilon(x^*)$ for $\varepsilon > 0$ such that:
 $f(y) \geq f(x^*) \quad \forall y \in B_\varepsilon(x^*)$



Global minimizer: $x^* \in \Omega$ is a global minimizer of \mathcal{P} if $f(y) \geq f(x^*) \quad \forall y \in \Omega$



Salient questions:

1. How to determine if x^* is a local minimizer?
2. How many local minima are there?
3. Is a local minimizer the global one?

Categories:

- Convex vs. non-convex



- Smooth vs. non-smooth  
 - Continuous vs. discrete $x \in \mathbb{R}^n$ $x \in \mathbb{Z}^n$
 - Constrained vs. unconstrained
 - Deterministic vs. stochastic
-

Today's focus: smooth and unconstrained

\mathcal{P}

$$\min_x f(x)$$

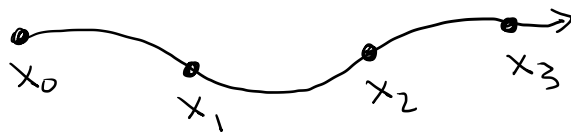
$$x \in \mathbb{R}^n \text{ (unconstrained)}$$

$$f \in \mathcal{C}^2 \text{ (smooth)}$$

Comment on
notation:

we've used

subscripts to denote "time"



we'll now use superscript to denote iteration:

$$x^{(0)} \xrightarrow{\text{gradient step}} x^{(1)} \xrightarrow{\text{gradient step}} x^{(2)}$$

How does our definition of a local minimizer
pertain to solving \mathcal{P} ?

we want $f(x^*)$ such that $f(x^* + \delta x) \geq f(x^*)$ "close" to $x^* \rightarrow$

$$f(x^* + \delta x) \approx f(x^*) + \nabla f(x^*)^T \delta x$$

$$\rightarrow f(x^* + \delta x) - f(x^*)$$

$$= f(x^*) + \nabla f(x^*)^T \delta x - f(x^*)$$

$$= \nabla f(x^*)^T \delta x \geq 0$$

we want this to be true $\forall \delta x \rightarrow \nabla f(x^*) = 0$

How to solve for this? Use iterative approach

pseudo-code: given $x^{(k)}$
$$x^{(k+1)} = x^{(k)} - \alpha \nabla f(x^{(k)})$$

if $\|\nabla f(x^{(k)})\|_2 \leq \epsilon_{\text{tol}}$
 terminate

α is the step size or learning rate

→ two ways we'll discuss today:

1. Newton's method
2. Backtracking line search

1. Newton's method: use 2nd-order Taylor expansion

$$\begin{aligned} f(\bar{x} + \delta x) &\approx f(\bar{x}) + \nabla f(\bar{x})^T \delta x + \frac{1}{2} \delta x^T \underbrace{\nabla_{xx} f(\bar{x})}_H \delta x \\ &= f(\bar{x}) + \nabla f(\bar{x})^T \delta x + \frac{1}{2} \delta x^T H \delta x := \bar{f} \end{aligned}$$

$$\rightarrow \nabla_x \mathcal{F} = \nabla f(\bar{x}) + H \delta x = 0$$

$$\rightarrow \delta x = -H^{-1} \nabla f(\bar{x}) = -(\nabla_{xx} f(\bar{x}))^{-1} \nabla f(\bar{x})$$

$$\alpha = (\nabla_{xx} f(\bar{x}))^{-1}$$

\rightarrow if $f(x)$ is quadratic, Newton method will converge in one iteration.

Since Newton method computes inverse of $\nabla_{xx} f(\bar{x})$ requires $\Theta(n^3)$ flops \rightarrow intractable for large n

2. backtracking line search: pick largest

$\alpha > 0$ such that

$$f(\bar{x} - \alpha \nabla f(\bar{x})) < f(\bar{x})$$

two stronger alternatives include:

Wolfe conditions: if $d^{(k)}$ is descent direction:

$$1. \quad f(x^{(k)} + \alpha d^{(k)}) \leq f(x^{(k)}) + c_1 \alpha d^{(k)T} \nabla f(x^{(k)})$$

α
jo
-tion

Armijo
condi

$$2. \quad -d^{(k)T} \nabla f(x^{(k)} + \alpha d^{(k)}) \leq -c_2 d^{(k)T} \nabla f(x^{(k)})$$

curvature
condition

Strong Wolfe conditions:

includes preceding two conditions (Armijo
+ curvature) and adds:

$$3. \quad |d^{(k)T} \nabla f(x^{(k)} + \alpha d^{(k)})| \leq c_2 |d^{(k)T} \nabla f(x^{(k)})|$$

Going back to \mathcal{P} , pseudo-code:

given $x^{(0)}$

for $i = 1 \rightarrow N_{\max}$

$$d^{(k)} = -\nabla_x f(x^{(k-1)}) \quad // \text{descent direction}$$

choose α \rightarrow LineSearch($x^{(k-1)}, \nabla_x f(x^{(k-1)})$)
 \rightarrow Newton($\nabla_x f(x^{(k-1)}), \nabla_{xx} f(x^{(k-1)})$)

$$x^{(k)} = x^{(k-1)} + \alpha d^{(k)}$$

$$\text{if } \|d^{(k)}\| \leq \varepsilon$$

break