Nonlinear Systems II: Multiple Variables

CS 205A:
Mathematical Methods for Robotics, Vision, and Graphics

Doug James (and Justin Solomon)

Announcements

- Midterm graded
- Discussed in Friday’s Section (see SCPD video)

Today's Root-Finding Problems

$$f : \mathbb{R}^n \to \mathbb{R}^m$$

One "Easy" Instance

$$f(\vec{x}) = A\vec{x} - \vec{b}$$

Usual Assumption

For $f : \mathbb{R}^n \to \mathbb{R}^m$, assume

$$n \geq m.$$

Examples (whiteboard)

Jacobian

$$(Df)_{ij} \equiv \frac{\partial f_i}{\partial x_j}$$

Jacobian

$$(Df)_{ij} \equiv \frac{\partial f_i}{\partial x_j}$$

How big is Df for $f : \mathbb{R}^n \to \mathbb{R}^m$?

First-Order Approximation of $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\vec{x}) \approx f(\vec{x}_k) + Df(\vec{x}_k) \cdot (\vec{x} - \vec{x}_k)$$

First-Order Approximation of $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\vec{x}) \approx f(\vec{x}_k) + Df(\vec{x}_k) \cdot (\vec{x} - \vec{x}_k)$$

Newton's Method:

$$\vec{x}_{k+1} = \vec{x}_k - [Df(\vec{x}_k)]^{-1} f(\vec{x}_k)$$

First-Order Approximation of $f : \mathbb{R}^n \to \mathbb{R}^n$

$$f(\vec{x}) \approx f(\vec{x}_k) + Df(\vec{x}_k) \cdot (\vec{x} - \vec{x}_k)$$

Newton's Method:

$$\vec{x}_{k+1} = \vec{x}_k - [Df(\vec{x}_k)]^{-1} f(\vec{x}_k)$$

Review: Do we explicitly compute $[Df(\vec{x}_k)]^{-1}$?

Convergence Sketch

1. $\vec{x}_{k+1} = g(\vec{x}_k)$ converges when the maximum-magnitude eigenvalue of Dg is less than 1

2. Extend observations about (quadratic) convergence in multiple dimensions

Common Examples

On whiteboard:

1. Implicit integration ($n = m$)
2. Projecting onto constraints ($n > m$)
   - E.g., Robotics (inverse kinematics)

Two Problems

1. Differentiation is hard

Two Problems

1. Differentiation is hard

2. $Df(\vec{x}_k)$ changes every iteration

Extend Secant Method?

Extend Secant Method?

Direct extensions are **not obvious!**

Observation: Directional Derivative

$$D_{\vec{v}}f = Df \cdot \vec{v}$$

Secant-Like Approximation

$$J \cdot (\vec{x}_k - \vec{x}_{k-1}) \approx f(\vec{x}_k) - f(\vec{x}_{k-1})$$

$$\text{where } J \approx Df(\vec{x}_k)$$

Secant-Like Approximation

$$J \cdot (\vec{x}_k - \vec{x}_{k-1}) \approx f(\vec{x}_k) - f(\vec{x}_{k-1})$$

where $J \approx Df(\vec{x}_k)$

“Broyden’s Method”

Broyden's Method: Outline

- Maintain current iterate $\vec{x}_k$ and approximation J_k of Jacobian near $\vec{x}_k$
- Update $\vec{x}_k$ using Newton-like step
- Update J_k using secant-like formula

Deriving the Broyden Step

$$\begin{aligned}\text{minimize}_{J_k}\ & \|J_k - J_{k-1}\|_{\text{Fro}}^2 \\ \text{such that }& J_k \cdot (\vec{x}_k - \vec{x}_{k-1}) = f(\vec{x}_k) - f(\vec{x}_{k-1})\end{aligned}$$

Deriving the Broyden Step

$$\begin{aligned}\text{minimize}_{J_k}\ & \|J_k - J_{k-1}\|^2_{\text{Fro}}\\ \text{such that } & J_k \cdot (\vec{x}_k - \vec{x}_{k-1}) = f(\vec{x}_k) - f(\vec{x}_{k-1})\end{aligned}$$

$$J_k = J_{k-1} + \frac{(f(\vec{x}_k) - f(\vec{x}_{k-1}) - J_{k-1}\cdot\Delta\vec{x})}{\|\vec{x}_k - \vec{x}_{k-1}\|_2^2}(\Delta\vec{x})^\top$$

The Newton Step

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

Implementation Details

- How to initialize J_0?

Implementation Details

- How to initialize J_0?
  Common choice: $J_0 \equiv I$

Implementation Details

- How to initialize J_0?
  Common choice: $J_0 \equiv I$

- Long "memory"—bad approximation?

Implementation Details

- How to initialize J_0?
  Common choice: $J_0 \equiv I$

- Long "memory"—bad approximation?
  Limited-memory methods

Implementation Details

- How to initialize J_0?
  Common choice: $J_0 \equiv I$

- Long "memory"—bad approximation?
  Limited-memory methods

- Still have to invert J_k in each step!

Revisiting the Broyden Step

$$J_k = J_{k-1} + \frac{(f(\vec{x}_k) - f(\vec{x}_{k-1}) - J_{k-1} \cdot \Delta\vec{x})}{\|\vec{x}_k - \vec{x}_{k-1}\|_2^2} (\Delta\vec{x})^\top$$

Revisiting the Broyden Step

$$J_k = J_{k-1} + \frac{(f(\vec{x}_k) - f(\vec{x}_{k-1}) - J_{k-1} \cdot \Delta\vec{x})}{\|\vec{x}_k - \vec{x}_{k-1}\|_2^2} (\Delta\vec{x})^\top$$

Simpler form:

$$J_k = J_{k-1} + \vec{u}_k \vec{v}_k^\top$$

Sherman-Morrison Formula

$$(A+\vec{u}\vec{v}^\top)^{-1} = A^{-1} - \frac{A^{-1}\vec{u}\vec{v}^\top A^{-1}}{1+\vec{v}^\top A^{-1}\vec{u}}$$

Homework: Check this.

Broyden Without Inversion

- Start with a J_0 for which you know J_0^{-1} (e.g. identity)
- Update J_0^{-1} directly!

Broyden Without Inversion

- Start with a J_0 for which you know J_0^{-1} (e.g. identity)
- Update J_0^{-1} directly!

Question: Limited-memory strategy?

Broyden Without Inversion

- Start with a J_0 for which you know J_0^{-1} (e.g. identity)
- Update J_0^{-1} directly!

Question: Limited-memory strategy?
Question: Large-scale strategy?

Automatic Differentiation

$$\left(x, \frac{dx}{dt}\right); \left(y, \frac{dy}{dt}\right) \mapsto$$

$$\left(x+y, \frac{dx}{dt}+\frac{dy}{dt}\right)$$

$$\left(xy, \frac{dx}{dt}y + x\frac{dy}{dt}\right) \qquad \left(\frac{x}{y}, \frac{y\frac{dx}{dt}+x\frac{dy}{dt}}{y^2}\right) \quad \cdots$$

JuliaDiff

- http://www.juliadiff.org
- Automatic differentiation of Julia code
- Generates Julia code for the derivative function
- ForwardDiff: Forward mode of automatic differentiation
- https://github.com/JuliaDiff/ForwardDiff.jl

▸ Next