Gauss-Newton: Observations?

$f(x) = \|r(x)\|_2$

$r(x) = y - a(x)$

$\varphi(x) = \frac{1}{2} f^2(x) \Rightarrow \nabla\varphi = J_r(x)^T r(x)$

$H_\varphi(x) = J_r^T J_r + \dots$

Demo: Gauss-Newton [cleared]

Observations?

- local conv
- slower than Newton (approximates Newton steps).

Gauss-Newton: $J^T J s = -\nabla\varphi$ (by linear least squares)

Levenberg-Marquardt

If Gauss-Newton on its own is poorly, conditioned, can try Levenberg-Marquardt:

G-N: $J_r(x_k)^T J_r(x_k) s_k = -J_r(x_k)^T r(x_k)$

LM: $(J_r(x_k)^T J_r(x_k) + \underbrace{\mu_k I}_{\text{"regularization"}}) s_k = \cdots$

$$\Downarrow$$

$$\begin{bmatrix} J_r(x_k) \\ \sqrt{\mu_k} \end{bmatrix} s_k \cong \begin{bmatrix} -r(x_k) \\ 0 \end{bmatrix}$$

Constrained Optimization: Problem Setup

Want $\boldsymbol{x}^*$ so that

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}) = 0$$

"Feasible"

No inequality constraints just yet. This is *equality-constrained optimization*. Develop a necessary condition for a minimum.

$\nabla f = 0$

$\vec{x} + \alpha \vec{s}$ has to be feasible

Constrained Optimization: Necessary Condition

$\nabla f \cdot \hat{s} \geq 0$ for any feasible search direction

↳ any feasible search direction would increase f

"In a puddle" $\Leftrightarrow$ not at bdry of $\{g=0\}$:

s and $-s$ are feasible search direction

$\Rightarrow \nabla f \cdot s \geq 0$ yields no extra information

$\Rightarrow \nabla f(x) = 0$ is a necessary condition

"At the boundary": Suppose $g(x)=0$.

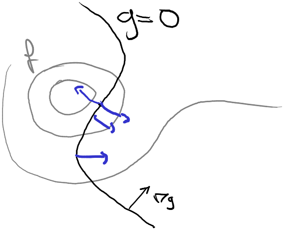
$-\nabla f(x) \in \text{rowspan}\ \partial g(x)$

necessary for min: "descent dir would violate # constraints"

Lagrange Multipliers

a constraint

$-\nabla f = J_g^T \cdot \vec{\lambda}$



Seen: Need $-\nabla f(\boldsymbol{x}) = J_{\boldsymbol{g}}^T \boldsymbol{\lambda}$ at the (constrained) optimum.

Idea: Turn constrained optimization problem for $\boldsymbol{x}$ into an *unconstrained* optimization problem for $(\boldsymbol{x}, \boldsymbol{\lambda})$. How?

$$\mathcal{L}(\vec{x}, \vec{\lambda}) = f(x) + \lambda^T g(x)$$

Lagrange Multipliers: Development

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) := f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{x}).$$

$$0 = \nabla \mathcal{L} = \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\lambda \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f + J_g^T \lambda \\ g(x) \end{bmatrix} = 0$$

matches our nec. cond.

= Newton applied to the Lagrangian

Demo: Sequential Quadratic Programming [cleared]

Inequality-Constrained Optimization

$g(x) \leq 0$

$-g(x) \leq 0$

Want $\boldsymbol{x}^*$ so that

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{g}(\boldsymbol{x}) = 0 \quad \text{and} \quad \underline{\boldsymbol{h}(\boldsymbol{x}) \leqslant 0}$$

This is *inequality-constrained optimization*. Develop a necessary condition for a minimum.

$$-\nabla f(x) = J_g^T \lambda_1$$

$$-\nabla f(x) = J_h^T \lambda_2, \quad \lambda_2 \geq 0$$

$$L(x, \lambda_1, \lambda_2) := f(x) + \lambda_1^T g(x) + \underline{\lambda_2^T} h(x)$$

- $h_i(x^*) = 0 \Rightarrow$ "active". $\lambda_{2i} > 0$
- $h_i(x^*) < 0 \Rightarrow$ "inactive". $\lambda_{2i} = 0$

Inequality-Constrained Optimization (cont'd)

Develop a set of necessary conditions for a minimum.

Karush-Kuhn-Tucker (KKT)

$$\nabla_x L(x^*, \lambda_1^*, \lambda_2^*) = 0 \quad (*)$$

$$g(x^*) = 0 \quad (*)$$

$$h(x^*) \leq 0$$

$$\lambda_2 \geq 0$$

$$h(x^*) \cdot \lambda_2 = 0 \quad (*)$$

"complementary condition"

Outline

Interpolation: Setup

Given: $(x_i)_{i=1}^N$, $(y_i)_{i=1}^N$
Wanted: Function f so that $f(x_i) = y_i$

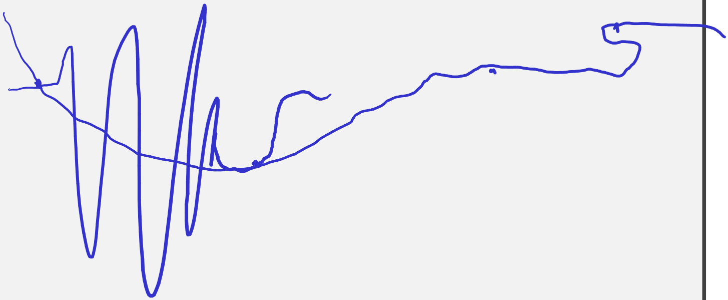
How is this not the same as function fitting? (from least squares)

f needs to hit (x_i, y_i) exactly

Interpolation: Setup (II)

Given: $(x_i)_{i=1}^N$, $(y_i)_{i=1}^N$
Wanted: Function f so that $f(x_i) = y_i$

Does this problem have a unique answer?



Interpolation: Importance

Why is interpolation important?

- Allows us to represent functions
- do calc on those Functions

Making the Interpolation Problem Unique

$$f(x) = \sum_{j=1}^{N_{func}} \alpha_j \varphi_j(x)$$

$$\underline{\underline{y}}_i = f(x_i) = \sum_{j=1}^{N_{func}} \underline{\underline{\alpha_j}} \underbrace{\varphi_j(x_i)}$$

$$\underline{\underline{V}}\alpha = y$$

(generalized) Vandermonde matrix

$$V_{ij} = \varphi_j(x_i)$$

Existence/Sensitivity

Solution to the interpolation problem: Existence? Uniqueness?

$V\alpha = y$ (linear algebra)

Sensitivity?

$$\max_{x\in[a,b]} |f(x)| \leq \Lambda \|y\|_\infty$$

Λ: Lebesgue constant

$\Lambda = \Lambda(n, x_i)$, same for all polynomial bases

Modes and Nodes (aka Functions and Points)

Both function basis and point set are under our control. What do we pick?

Ideas for basis functions:

- Monomials $1, x, x^2, x^3, x^4, \ldots$
- Functions that make $V = I \to$ 'Lagrange basis'
- Functions that make V triangular $\to$ 'Newton basis'
- *Splines* (piecewise polynomials)
- *Orthogonal polynomials*
- Sines and cosines
- 'Bumps' (*'Radial Basis Functions'*)

Ideas for points:

- Equispaced
- *'Edge-Clustered'* (so-called Chebyshev/Gauss/... nodes)

Specific issues:

- Why *not* monomials on equispaced points?
  Demo: Monomial interpolation [cleared]
- Why not equispaced?
  Demo: Choice of Nodes for Polynomial Interpolation [cleared]

Lagrange Interpolation

Find a basis so that $V = I$, i.e.

$$\varphi_j(x_i) = \begin{cases} 1 & i = j, \\ 0 & \text{otherwise.} \end{cases}$$