

## 13. Nonlinear least squares

- nonlinear least squares
- Gauss-Newton method
- Levenberg-Marquardt method
- nonlinear data fitting

## Nonlinear least squares

$$\text{minimize } \sum_{i=1}^m f_i(x)^2 = \|f(x)\|^2$$

- $x$  is variable,  $f_1(x), \dots, f_m(x)$  are *residuals*
- $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is vector residual with components  $f_i(x)$ :

$$f(x) = (f_1(x), f_2(x), \dots, f_m(x))$$

- objective function is  $\|f(x)\|^2$
- problem reduces to (linear) least squares if  $f(x) = Ax - b$
- solution approximate or solves the set of nonlinear equations  $f(x) = 0$

## Example: Location from range measurements

- 3-vector  $x$  is position in 3-D, which we will estimate
- range measurements give (noisy) distance to known locations

$$\rho_i = \|x - a_i\| + v_i, \quad i = 1, \dots, m$$

$a_i$  are known locations,  $v_i$  are noises

- least squares location estimation: choose  $\hat{x}$  that minimizes

$$\sum_{i=1}^m (\|x - a_i\| - \rho_i)^2$$

- GPS works like this

## Gradient of nonlinear least squares cost

$$g(x) = \|f(x)\|^2 = \sum_{i=1}^m f_i(x)^2$$

- first derivative of  $g$  with respect to  $x_j$ :

$$\frac{\partial g}{\partial x_j}(z) = 2 \sum_{i=1}^m f_i(z) \frac{\partial f_i}{\partial x_j}(z)$$

- gradient of  $g$  at  $z$ :

$$\nabla g(z) = \begin{bmatrix} \frac{\partial g}{\partial x_1}(z) \\ \vdots \\ \frac{\partial g}{\partial x_n}(z) \end{bmatrix} = 2 \sum_{i=1}^m f_i(z) \nabla f_i(z) = 2 Df(z)^T f(z)$$

## Optimality condition

$$\text{minimize } g(x) = \sum_{i=1}^m f_i(x)^2$$

**necessary condition for optimality:** if  $x$  minimizes  $g(x)$  then it must satisfy

$$\nabla g(x) = 2Df(x)^T f(x) = 0$$

- this generalizes the normal equations: if  $f(x) = Ax - b$ , then  $Df(x) = A$  and

$$\nabla g(x) = 2A^T(Ax - b)$$

- for general  $f$ , the condition  $\nabla g(x) = 0$  is not sufficient for optimality

## Difficulty of solving nonlinear least squares problem

- solving nonlinear equations or nonlinear least squares problem is (in general) much harder than solving linear equations
- even determining if a solution exists is hard
- so we will use heuristic algorithms
- not guaranteed to always work
- but often work well in practice (like  $k$ -means)

# Outline

- nonlinear least squares
- **Gauss-Newton method**
- Levenberg-Marquardt method
- nonlinear data fitting

## Linear least square approximation at each iteration

$$\text{minimize } g(x) = \sum_{i=1}^m f_i(x)^2$$

- $x^{(k)}$  is estimate of a solution at time  $k$
- $\hat{f}(x; x^{(k)})$  is first order Taylor approximation of  $f$  around  $x^{(k)}$ :

$$\hat{f}(x; x^{(k)}) = f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})$$

this is a good approximation if  $x$  near  $x^{(k)}$  ( $\|x - x^{(k)}\|$  is small)

- Gauss-Newton method produces new estimate  $x^{(k+1)}$  that solves the problem

$$\text{minimize } \|\hat{f}(x; x^{(k)})\|^2 = \|f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})\|^2$$

- the above problem is a linear least-squares problem with

$$A = Df(x^{(k)}), \quad b = Df(x^{(k)})x^{(k)} - f(x^{(k)})$$



## Gauss-Newton method

setting  $x^{(k+1)}$  to be the solution of the previous problem, we have

$$\begin{aligned}x^{(k+1)} &= (A^T A)^{-1} A^T b \\&= (Df(x^{(k)})^T Df(x^{(k)}))^{-1} Df(x^{(k)})^T (Df(x^{(k)})x^{(k)} - f(x^{(k)})) \\&= x^{(k)} - (Df(x^{(k)})^T Df(x^{(k)}))^{-1} Df(x^{(k)})^T f(x^{(k)})\end{aligned}$$

- assumes that  $A = Df(x^{(k)})$  has linearly independent columns
- if converged (i.e.,  $x^{(k+1)} = x^{(k)}$ ) then

$$Df(x^{(k)})^T f(x^{(k)}) = 0$$

hence  $x^{(k)}$  satisfies the optimality condition since gradient is  $2Df(x)^T f(x)$

## Gauss-Newton algorithm

---

**given** a starting point  $x^{(1)}$  and solution tolerance  $\epsilon$

**repeat for**  $k \geq 0$ :

1. evaluate  $Df(x^{(k)}) = (\nabla f_1(x^{(k)})^T, \dots, \nabla f_m(x^{(k)})^T)$

2. set

$$x^{(k+1)} = x^{(k)} - (Df(x^{(k)})^T Df(x^{(k)}))^{-1} Df(x^{(k)})^T f(x^{(k)})$$

**if** stopping criteria holds, stop and output  $x^{(k+1)}$

---

### Stopping criteria

$$\|f(x^{(k)})\|^2 \leq \epsilon, \quad \|Df(x^{(k)})^T f(x^{(k)})\| \leq \epsilon, \quad \|x^{(k+1)} - x^{(k)}\| \leq \epsilon$$

- if  $x^{(k+1)} = x^{(k)}$ , then  $x^{(k)}$  satisfies the optimality condition
- this does not mean that  $x^{(k)}$  is a good solution
- it is common to run the algorithm from different starting points and choose the best solution of these multiple runs

## Issues with Gauss-Newton method

- approximation

$$\|f(x)\|^2 \approx \|\hat{f}(x; x^{(k)})\|^2$$

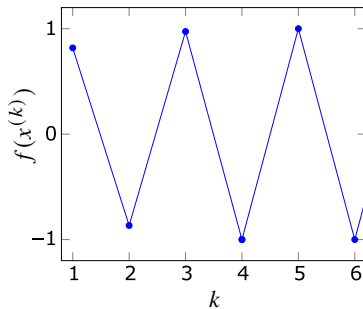
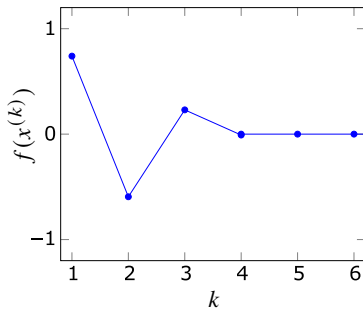
holds when  $x$  near  $x^{(k)}$

- when  $x^{(k+1)}$  is not near  $x^{(k)}$ , the affine approximation will not be accurate
- so the algorithm may fail or diverge ( $\|f(x^{(k+1)})\| > \|f(x^{(k)})\|$ )
- a second major issue is that columns of the matrix  $Df(x^{(k)})$  may not always be linearly independent; in this case, the next iterate is not defined

## Example

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- starting point  $x^{(1)} = 0.9$ : converges very rapidly to  $x^\star = 0$
- starting point  $x^{(1)} = 1.1$ : does not converge



## Relation to Newton method for nonlinear equations

- Gauss-Newton update

$$x^{(k+1)} = x^{(k)} - \left( Df(x^{(k)})^T Df(x^{(k)}) \right)^{-1} Df(x^{(k)})^T f(x^{(k)})$$

- if  $m = n$ , then  $Df(x)$  is square and this is the Newton update

$$x^{(k+1)} = x^{(k)} - Df(x^{(k)})^{-1} f(x^{(k)})$$

## Relation to Newton method for unconstrained minimization

$$g(x) = \|f(x)\|^2 = \sum_{i=1}^m f_i(x)^2$$

- gradient:

$$\nabla g(x) = 2 \sum_{i=1}^m f_i(x) \nabla f_i(x) = 2Df(x)^T f(x)$$

- second derivatives:

$$\frac{\partial^2 g}{\partial x_j \partial x_k}(x) = 2 \sum_{i=1}^m \left( \frac{\partial f_i}{\partial x_j}(x) \frac{\partial f_i}{\partial x_k}(x) + f_i(x) \frac{\partial^2 f_i}{\partial x_j \partial x_k}(x) \right)$$

- Hessian

$$\nabla^2 g(x) = 2Df(x)^T Df(x) + 2 \sum_{i=1}^m f_i(x) \nabla^2 f_i(x)$$

## Newton and Gauss-Newton steps

(Undamped) Newton step at  $x = x^{(k)}$ :

$$\begin{aligned}v_{\text{nt}} &= -\nabla^2 g(x)^{-1} \nabla g(x) \\&= -\left(Df(x)^T Df(x) + \sum_{i=1}^m f_i(x) \nabla^2 f_i(x)\right)^{-1} Df(x)^T f(x)\end{aligned}$$

Gauss-Newton step at  $x = x^{(k)}$ :

$$v_{\text{gn}} = -\left(Df(x)^T Df(x)\right)^{-1} Df(x)^T f(x)$$

- can be written as  $v_{\text{gn}} = -H_{\text{gn}}^{-1} \nabla g(x)$  where  $H_{\text{gn}} = Df(x)^T Df(x)$
- $H_{\text{gn}}$  is the Hessian without the term  $\sum_i f_i(x) \nabla^2 f_i(x)$

# Comparison

## Newton step

- requires second derivatives of  $f$
- not always a descent direction ( $\nabla^2 g(x)$  is not necessarily positive definite)
- fast convergence near local minimum

## Gauss-Newton step

- does not require second derivatives
- a descent direction (if columns of  $Df(x)$  are linearly independent):

$$\nabla g(x)^T v_{\text{gn}} = -2v_{\text{gn}}^T Df(x)^T Df(x) v_{\text{gn}} < 0 \quad \text{if } v_{\text{gn}} \neq 0$$

- local convergence to  $x^\star$  is similar to Newton method if

$$\sum_{i=1}^m f_i(x^\star) \nabla^2 f_i(x^\star)$$

is small (for each  $i$ ,  $f_i(x^\star)$  is small or  $f_i$  is nearly affine around  $x^\star$ )



# Outline

- nonlinear least squares
- Gauss-Newton method
- **Levenberg-Marquardt method**
- nonlinear data fitting

## Regularized approximate problem

ensure  $x$  is close to  $x^{(k)}$  by regularization

$$\text{minimize} \quad \|f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})\|^2 + \lambda^{(k)} \|x - x^{(k)}\|^2$$

- regularization parameter  $\lambda^{(k)}$  controls how close  $x^{(k+1)}$  is to  $x^{(k)}$
- regularization fixes invertibility issue of Gauss-Newton (no condition on  $Df(x)$ )
- the above problem can be rewritten as

$$\text{minimize} \quad \left\| \begin{bmatrix} Df(x^{(k)}) \\ \sqrt{\lambda^{(k)}} I \end{bmatrix} x - \begin{bmatrix} Df(x^{(k)})x^{(k)} - f(x^{(k)}) \\ \sqrt{\lambda^{(k)}} x^{(k)} \end{bmatrix} \right\|^2$$

this is just a least-squares problem with objective  $\|Ax - b\|^2$  where

$$A = \begin{bmatrix} Df(x^{(k)}) \\ \sqrt{\lambda^{(k)}} I \end{bmatrix}, \quad b = \begin{bmatrix} Df(x^{(k)})x^{(k)} - f(x^{(k)}) \\ \sqrt{\lambda^{(k)}} x^{(k)} \end{bmatrix}$$

the solution is

$$x^{(k+1)} = x^{(k)} - (Df(x^{(k)})^T Df(x^{(k)}) + \lambda^{(k)} I)^{-1} Df(x^{(k)})^T f(x^{(k)})$$

we see  $x^{(k+1)} = x^{(k)}$  only if optimality condition hold  $Df(x^{(k)})^T f(x^{(k)})$

### Updating $\lambda^{(k)}$

- if  $\lambda^{(k)}$  is very small, then  $x^{(k+1)}$  can be far from  $x^{(k)}$ , and the method may fail
- if  $\lambda^{(k)}$  is large enough, then  $x^{(k+1)}$  becomes close to  $x^{(k)}$  and the affine approximation will be accurate enough
- a simple way to update  $\lambda^{(k)}$  is to check whether

$$\|f(x^{(k+1)})\|^2 < \|f(x^{(k)})\|^2$$

if so, then we can decrease  $\lambda^{(k+1)}$ ; otherwise, we increase  $\lambda^{(k+1)}$

# Levenberg-Marquardt algorithm

---

**given** a starting point  $x^{(1)}$ , solution tolerance  $\epsilon$ , and  $\lambda^{(1)} > 0$

**repeat for**  $k \geq 0$

1. evaluate  $Df(x^{(k)}) = (\nabla f_1(x^{(k)})^T, \dots, \nabla f_m(x^{(k)})^T)$
2. update

$$x^{(k+1)} = x^{(k)} - (Df(x^{(k)})^T Df(x^{(k)}) + \lambda^{(k)} I)^{-1} Df(x^{(k)})^T f(x^{(k)})$$

**if** stopping criteria holds, stop and output  $x^{(k+1)}$

3. **if**  $\|f(x^{(k+1)})\|^2 < \|f(x^{(k)})\|^2$ , then decrease  $\lambda^{(k+1)}$  (e.g.,  $\lambda^{(k+1)} = 0.8\lambda^{(k)}$ );  
otherwise, increase  $\lambda^{(k+1)}$  (e.g.,  $\lambda^{(k+1)} = 2\lambda^{(k)}$ ) and keep  $x^{(k)} = x^{(k+1)}$
- 

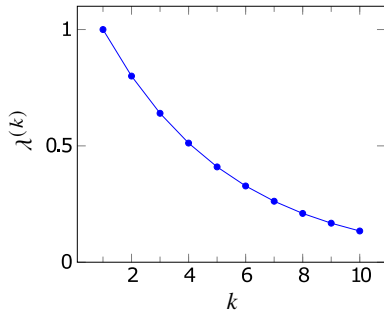
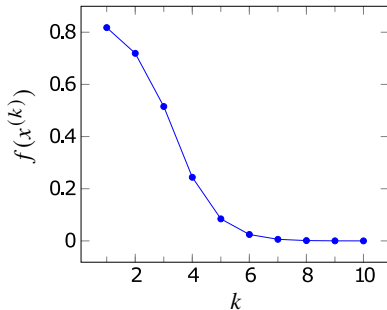
## Stopping criteria

$$\|f(x^{(k)})\|^2 \leq \epsilon, \quad \|Df(x^{(k)})^T f(x^{(k)})\| \leq \epsilon, \quad \|x^{(k+1)} - x^{(k)}\| \leq \epsilon$$

## Example

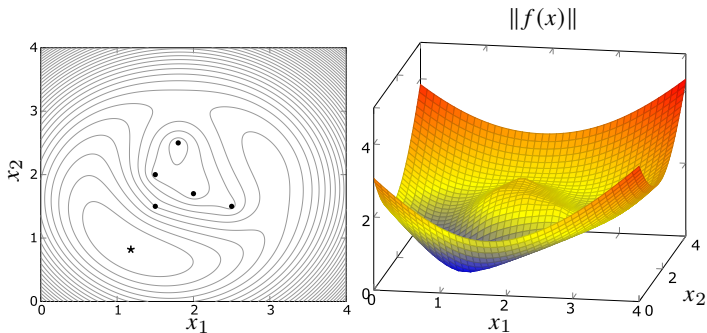
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- we saw Gauss-Newton does not converge starting at  $x^{(1)} = 1.1$
- for Levenberg-Marquardt starting at  $x^{(1)} = 1.1$  and  $\lambda^{(1)} = 1$  converges

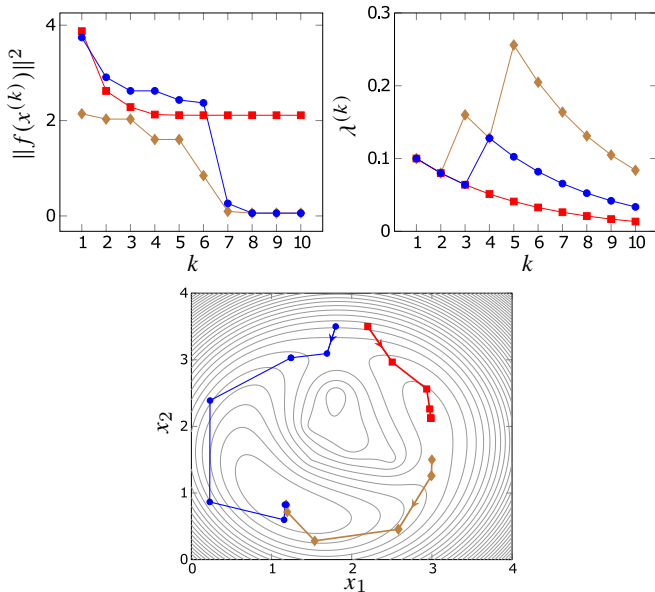


## Example: Location from range measurements

- range to 5 points (blue circles)
- red square shows  $\hat{x}$



## Levenberg-Marquardt from three initial points



# Outline

- nonlinear least squares
- Gauss-Newton method
- Levenberg-Marquardt method
- **nonlinear data fitting**



## Nonlinear model fitting

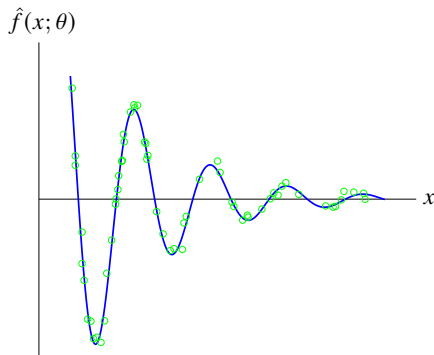
$$\text{minimize} \quad \sum_{i=1}^N (\hat{f}(x^{(i)}; \theta) - y^{(i)})^2$$

- $x^{(1)}, \dots, x^{(N)}$  are feature vectors and  $y^{(1)}, \dots, y^{(N)}$  are associated outcomes
- model  $\hat{f}(x; \theta)$  is parameterized by parameters  $\theta_1, \dots, \theta_p$
- this generalizes the linear in parameters model

$$\hat{f}(x; \theta) = \theta_1 f_1(x) + \dots + \theta_p f_p(x)$$

- here we allow  $\hat{f}(x, \theta)$  to be a nonlinear function of  $\theta$
- the minimization is over the model parameters  $\theta$

## Example

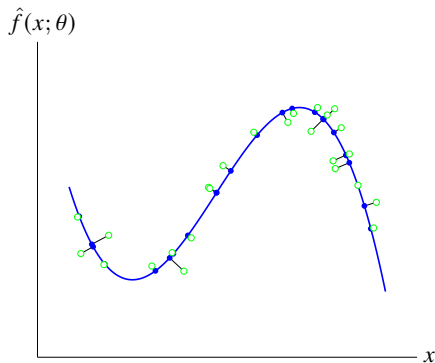


a nonlinear least squares problem with four variables  $\theta_1, \theta_2, \theta_3, \theta_4$ :

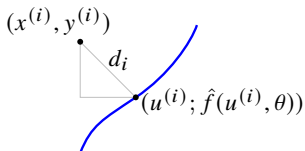
$$\text{minimize} \quad \sum_{i=1}^N \left( \theta_1 e^{\theta_2 x^{(i)}} \cos(\theta_3 x^{(i)} + \theta_4) - y^{(i)} \right)^2$$

## Orthogonal distance regression

- to fit model, minimize sum square distance of data points to graph
- example: orthogonal distance regression to cubic polynomial



## Nonlinear least squares formulation



$$d_i^2 = (\hat{f}(u^{(i)}, \theta) - y^{(i)})^2 + \|u^{(i)} - x^{(i)}\|^2$$

- linear in parameters model:  $\hat{f}(x; \theta) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$
- minimizing over  $(u^{(i)}, \theta)$  gives squared distance of  $(x^{(i)}, y^{(i)})$  to graph  $\hat{f}$

### Orthogonal distance regression

$$\text{minimize} \quad \sum_{i=1}^N \left( (\hat{f}(u^{(i)}; \theta) - y^{(i)})^2 + \|u^{(i)} - x^{(i)}\|^2 \right)$$

- optimization variables are model parameters  $\theta$  and  $N$  points  $u^{(i)}$
- $i$ th term is squared distance of data point  $(x^{(i)}, y^{(i)})$  to point  $(u^{(i)}, \hat{f}(u^{(i)}, \theta))$

# Classification

## Linear least squares classifier

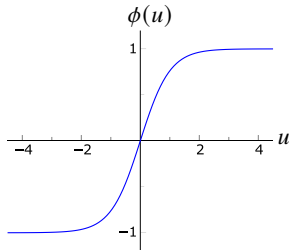
- data points  $(x^{(i)}, y^{(i)})$  where  $y^{(i)} \in \{-1, 1\}$
- classifier is  $\hat{f}(x) = \text{sign}(\tilde{f}(x))$  where  $\tilde{f}(x) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x)$
- $\theta$  is chosen by minimizing  $\sum_{i=1}^N (\tilde{f}(x_i) - y_i)^2$  (plus optionally regularization)

## Nonlinear least squares classifier

- choose  $\theta$  to minimize  $\sum_{i=1}^N (\text{sign}(\tilde{f}(x^{(i)})) - y^{(i)})^2$
- replace sign function with smooth approximation  $\phi$ , e.g., sigmoid function

$$\text{minimize} \quad \sum_{i=1}^N (\phi(\tilde{f}(x^{(i)})) - y^{(i)})^2$$

$$\phi(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$



## References and further readings

- S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*, Cambridge University Press, 2018.
- L. Vandenberghe. *EE133A lecture notes*, Univ. of California, Los Angeles.  
(<http://www.seas.ucla.edu/~vandenbe/ee133a.html>)