# 18. Gauss–Newton method

- definition and examples
- Gauss–Newton method
- Levenberg–Marquardt method
- separable nonlinear least squares

# Nonlinear least squares

$$\text{minimize} \quad g(x) = \sum_{i=1}^{m} f_i(x)^2 = \|f(x)\|_2^2$$

- $f : \mathbf{R}^n \to \mathbf{R}^m$ is a differentiable function $f(x) = (f_1(x), \ldots, f_m(x))$ of $n$-vector $x$
- in general, a nonconvex optimization problem
- linear least squares is special case with $f(x) = Ax - b$

$$x^\star = A^+ b, \qquad g(x^\star) = \|(I - AA^+)b\|_2^2 = b^T(I - AA^+)b$$

  $A^+$ is the pseudo-inverse: $A^+ = (A^T A)^{-1} A^T$ if $A$ has full column rank

- as in lecture 16 we denote the $m \times n$ Jacobian matrix of $f$ by $f'(x)$:

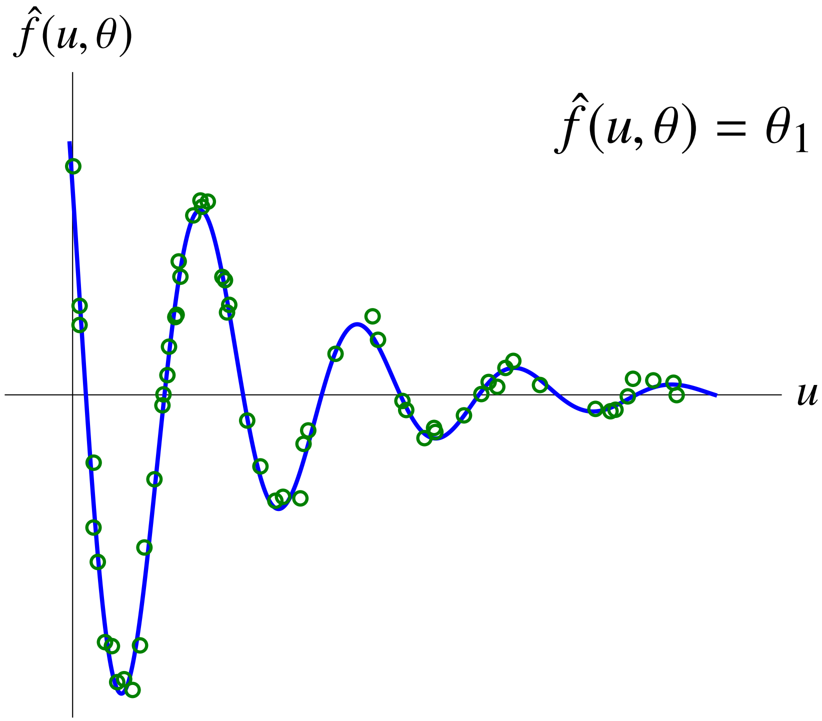
$$f'(x) = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_m(x)^T \end{bmatrix}$$

# Model fitting

$$\text{minimize} \quad \sum_{i=1}^{N} (\hat{f}(u^{(i)}, \theta) - v^{(i)})^2$$

- model $\hat{f}(u, \theta)$ depends on model parameters $\theta_1, \ldots, \theta_p$
- $(u^{(1)}, v^{(1)}), \ldots, (u^{(N)}, v^{(N)})$ are data points
- the minimization is over the model parameters $\theta$

**Example**



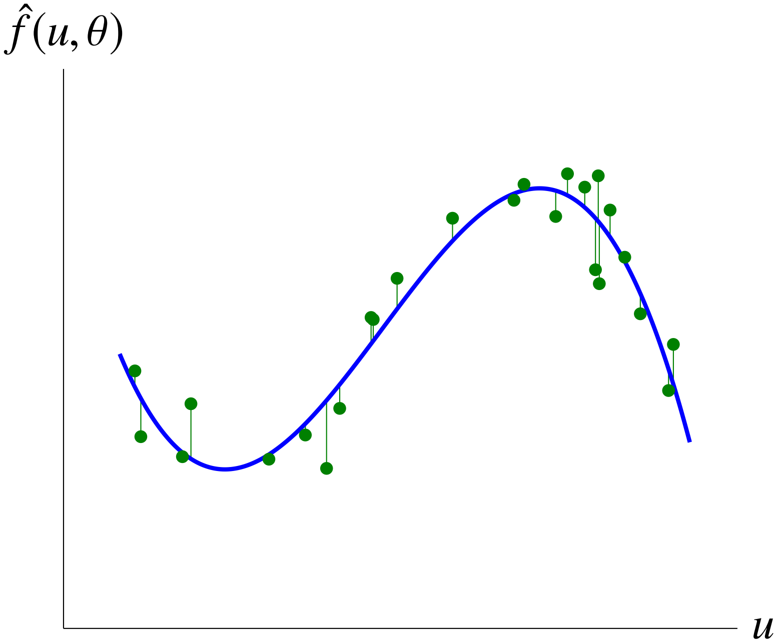

$$\hat{f}(u, \theta) = \theta_1 \exp(\theta_2 u) \cos(\theta_3 u + \theta_4)$$

# Orthogonal distance regression

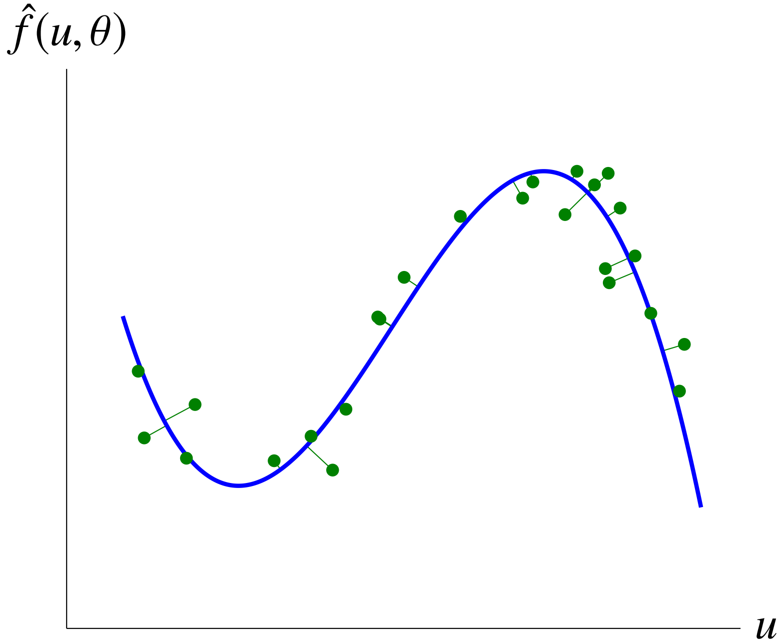
minimize the mean square distance of data points to graph of $\hat{f}(u, \theta)$

**Example:** orthogonal distance regression with cubic polynomial

$$\hat{f}(u, \theta) = \theta_1 + \theta_2 u + \theta_3 u^2 + \theta_4 u^3$$




standard least squares fit

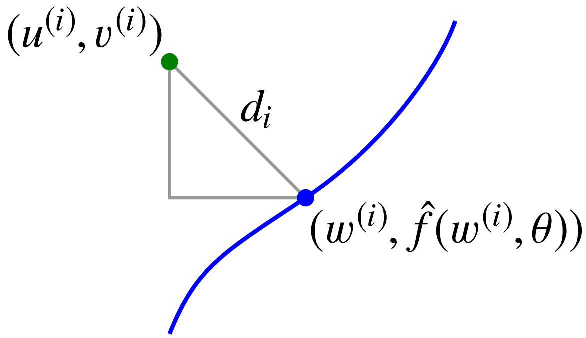



orthogonal distance fit

# Nonlinear least squares formulation

$$\text{minimize} \quad \sum_{i=1}^{N} \left( (\hat{f}(w^{(i)}, \theta) - v^{(i)})^2 + \|w^{(i)} - u^{(i)}\|_2^2 \right)$$

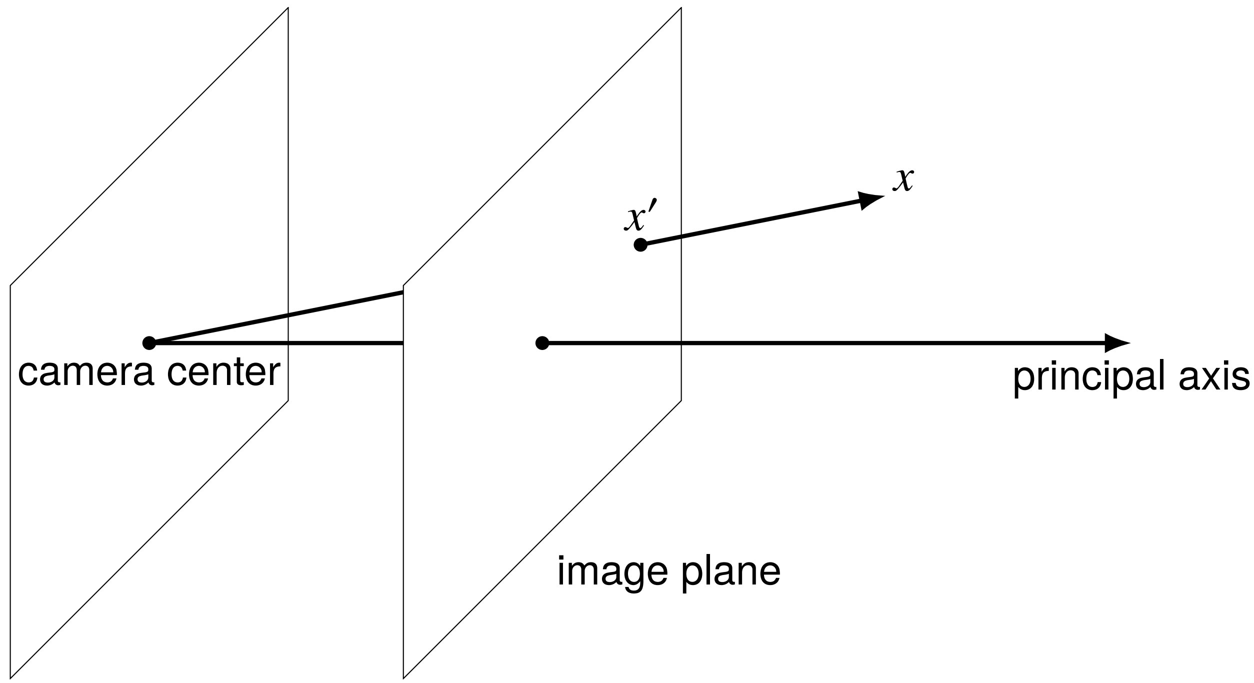
- optimization variables are model parameters $\theta$ and $N$ points $w^{(i)}$
- $i$th term is squared distance of data point $(u^{(i)}, v^{(i)})$ to point $(w^{(i)}, \hat{f}(w^{(i)}, \theta))$




$$d_i^2 = (\hat{f}(w^{(i)}, \theta) - v^{(i)})^2 + \|w^{(i)} - u^{(i)}\|_2^2$$

- minimizing $d_i^2$ over $w^{(i)}$ gives squared distance of $(u^{(i)}, v^{(i)})$ to graph
- minimizing $\sum_i d_i^2$ over $w^{(1)}, \ldots, w^{(N)}$ and $\theta$ minimizes mean squared distance

# Location from multiple camera views




**Camera model:** described by parameters $A \in \mathbf{R}^{2\times 3}$, $b \in \mathbf{R}^2$, $c \in \mathbf{R}^3$, $d \in \mathbf{R}$

- object at location $x \in \mathbf{R}^3$ creates image at location $x' \in \mathbf{R}^2$ in image plane

$$x' = \frac{1}{c^T x + d}(Ax + b)$$

  $c^T x + d > 0$ if object is in front of the camera

- $A$, $b$, $c$, $d$ characterize the camera, and its position and orientation

# Location from multiple camera views

- an object at location $x_{\text{ex}}$ is viewed by $l$ cameras (described by $A_i$, $b_i$, $c_i$, $d_i$)
- the image of the object in the image plane of camera $i$ is at location

$$y_i = \frac{1}{c_i^T x_{\text{ex}} + d_i}(A_i x_{\text{ex}} + b_i) + v_i$$

- $v_i$ is measurement or quantization error
- goal is to estimate 3-D location $x_{\text{ex}}$ from the $l$ observations $y_1, \ldots, y_l$

**Nonlinear least squares estimate**: compute estimate $\hat{x}$ by minimizing

$$\sum_{i=1}^{l} \left\| \frac{1}{c_i^T x + d_i}(A_i x + b_i) - y_i \right\|_2^2$$

# Outline

- definition and examples
- **Gauss–Newton method**
- Levenberg–Marquardt method
- separable nonlinear least squares

# Gauss–Newton method

$$\text{minimize} \quad \|f(x)\|_2^2 = \sum_{i=1}^{m} f_i(x)^2$$

start at some initial guess $x_0$, and repeat for $k = 1, 2, \ldots$:

- linearize $f$ around $x_k$:

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k)$$

- substitute affine approximation for $f$ in least squares problem:

$$\text{minimize} \quad \left\| f(x_k) + f'(x_k)(x - x_k) \right\|_2^2$$

- take the solution of this linear least squares problem as $x_{k+1}$

# Gauss–Newton update

least squares problem solved in iteration $k$:

$$\text{minimize} \quad \|f'(x_k)(x - x_k) + f(x_k)\|_2^2$$

- if $f'(x_k)$ has full column rank, solution is given by

$$\begin{aligned} x_{k+1} &= x_k - (f'(x_k)^T f'(x_k))^{-1} f'(x_k)^T f(x_k) \\ &= x_k - f'(x_k)^+ f(x_k) \end{aligned}$$

- Gauss–Newton step $v_k = x_{k+1} - x_k$ is the solution of the linear LS problem

$$\text{minimize} \quad \|f'(x_k)v + f(x_k)\|_2^2$$

- to improve convergence, can add line search and update $x_{k+1} = x_k + t_k v_k$

# Newton and Gauss–Newton steps

$$\text{minimize} \quad g(x) = \|f(x)\|_2^2 = \sum_{i=1}^{m} f_i(x)^2$$

**Newton step** at $x = x_k$:

$$\begin{aligned} v_{\text{nt}} &= -\nabla^2 g(x)^{-1} \nabla g(x) \\ &= -\left( f'(x)^T f'(x) + \sum_{i=1}^{m} f_i(x) \nabla^2 f_i(x) \right)^{-1} f'(x)^T f(x) \end{aligned}$$

**Gauss–Newton step** at $x = x_k$ (from previous page):

$$v_{\text{gn}} = -\left( f'(x)^T f'(x) \right)^{-1} f'(x)^T f(x)$$

- this can be written as $v_{\text{gn}} = -H^{-1} \nabla g(x)$ where $H = 2 f'(x)^T f'(x)$
- $H$ is the Hessian without the terms $f_i(x) \nabla^2 f_i(x)$

# 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: $H = 2f'(x)^T f'(x) \succ 0$ (if $f'(x)$ has full column rank)
- 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 (*e.g.*, $f(x^\star)$ is small, or $f$ is nearly affine around $x^\star$)

# Outline

- definition and examples
- Gauss–Newton method
- **Levenberg–Marquardt method**
- separable nonlinear least squares

# Levenberg–Marquardt method

addresses two difficulties in Gauss–Newton method:

- how to update $x_k$ when columns of $f'(x_k)$ are linearly dependent
- what to do when the Gauss–Newton update does not reduce $\|f(x)\|_2^2$

## Levenberg–Marquardt method

compute $x_{k+1}$ by solving a *regularized* least squares problem

$$\text{minimize} \quad \|f'(x_k)(x - x_k) + f(x_k)\|_2^2 + \lambda_k \|x - x_k\|_2^2$$

- second term forces $x$ to be close to $x_k$ where local approximation is accurate
- with $\lambda_k > 0$, always has a unique solution (no rank condition on $f'(x_k)$)
- a proximal point update with convexified cost function

# Levenberg–Marquardt update

regularized least squares problem solved in iteration $k$

$$\text{minimize} \quad \left\| f'(x_k)(x - x_k) + f(x_k) \right\|_2^2 + \lambda_k \|x - x_k\|_2^2$$

- solution is given by

$$x_{k+1} = x_k - \left( f'(x_k)^T f'(x_k) + \lambda_k I \right)^{-1} f'(x_k)^T f(x_k)$$

- Levenberg–Marquardt step $v_k = x_{k+1} - x_k$ is

$$\begin{aligned} v_k &= -\left( f'(x_k)^T f(x_k) + \lambda_k I \right)^{-1} f'(x_k)^T f(x_k) \\ &= -\frac{1}{2} \left( f'(x_k)^T f'(x_k) + \lambda_k I \right)^{-1} \nabla g(x_k) \end{aligned}$$

- for $\lambda_k = 0$ this is the Gauss–Newton step (if defined); for large $\lambda_k$,

$$v_k \approx -\frac{1}{2\lambda_k} \nabla g(x_k)$$

# Regularization parameter

several strategies for adapting $\lambda_k$ are possible; for example:

- at iteration $k$, compute the solution $v$ of

$$\text{minimize} \quad \left\| f'(x_k)v + f(x_k) \right\|_2^2 + \lambda_k \|v\|_2^2$$

- if $\|f(x_k + v)\|_2^2 < \|f(x_k)\|_2^2$, take $x_{k+1} = x_k + v$ and decrease $\lambda$
- otherwise, do not update $x$ (take $x_{k+1} = x_k$), but increase $\lambda$

## Some variations

- compare actual cost reduction with reduction predicted by linearized problem
- solve a least squares problem with trust region

$$\begin{array}{ll} \text{minimize} & \|f'(x_k)v + f(x_k)\|_2^2 \\ \text{subject to} & \|v\|_2 \leq \gamma \end{array}$$

# Summary: Levenberg–Marquardt method

choose $x_0$ and $\lambda_0$ and repeat for $k = 0, 1, \ldots$:

1. evaluate $f(x_k)$ and $A = f'(x_k)$
2. compute solution of regularized least squares problem:

$$\hat{x} = x_k - (A^T A + \lambda_k I)^{-1} A^T f(x_k)$$

3. define $x_{k+1}$ and $\lambda_{k+1}$ as follows:

$$\begin{cases} x_{k+1} = \hat{x} \text{ and } \lambda_{k+1} = \beta_1 \lambda_k & \text{if } \|f(\hat{x})\|_2^2 < \|f(x_k)\|_2^2 \\ x_{k+1} = x_k \text{ and } \lambda_{k+1} = \beta_2 \lambda_k & \text{otherwise} \end{cases}$$

- $\beta_1$, $\beta_2$ are constants with $0 < \beta_1 < 1 < \beta_2$
- terminate if $\nabla g(x_k) = 2A^T f(x_k)$ is sufficiently small

# Outline

- definition and examples
- Gauss–Newton method
- Levenberg–Marquardt method
- separable nonlinear least squares

# Separable nonlinear least squares

$$\text{minimize} \quad \|A(y)x - b(y)\|_2^2$$

- $A : \mathbf{R}^p \to \mathbf{R}^{m\times n}$ and $b : \mathbf{R}^p \to \mathbf{R}^m$ are differentiable functions
- variables are $x \in \mathbf{R}^m$ and $y \in \mathbf{R}^p$
- reduces to linear least squares if $A(y)$ and $b(y)$ are constant

**Example:** the separable structure is common in model fitting problems

$$\text{minimize} \quad \sum_{i=1}^{N} \left(\hat{f}(u^{(i)}, \theta) - v^{(i)}\right)^2$$

- model $\hat{f}$ is linear combination of parameterized basis functions: $\theta = (x, y)$ and

$$\hat{f}(u, \theta) = x_1 h_1(u, y) + \cdots + x_p h_p(u, y)$$

- variables are coefficients $x_1, \ldots, x_p$ and parameters $y$

# Derivative notation

$$f(x, y) = A(y)x - b(y)$$

- $y$ is a $p$-vector, $x$ is an $n$-vector, $A(y)$ is an $m \times n$ matrix
- we denote the rows of $A(y)$ by $a_i(y)^T$, with $a_i(y) \in \mathbf{R}^n$:

$$A(y) = \begin{bmatrix} a_1(y)^T \\ \vdots \\ a_m(y)^T \end{bmatrix}$$

- the Jacobian of $f(x, y)$ is the $m \times (n + p)$ matrix

$$f'(x, y) = \begin{bmatrix} A(y) & B(x, y) \end{bmatrix}, \qquad \text{where } B(x, y) = \begin{bmatrix} x^T a_1'(y) \\ \vdots \\ x^T a_m'(y) \end{bmatrix} + b'(y)$$

here $a_i'(y) \in \mathbf{R}^{n \times p}$ and $b'(y) \in \mathbf{R}^{m \times p}$ are the Jacobian matrices of $a_i$, $b$

# Gauss–Newton algorithm

$$\text{minimize} \quad \|f(x,y)\|_2^2 = \|A(y)x - b(y)\|_2^2$$

- in the Gauss–Newton algorithm we choose for $x_{k+1}$, $y_{k+1}$ the solution $x$, $y$ of

$$\text{minimize} \quad \left\| \begin{bmatrix} A(y_k) & B(x_k, y_k) \end{bmatrix} \begin{bmatrix} x \\ y - y_k \end{bmatrix} - b(y_k) \right\|_2^2$$

- if we eliminate $x$ in this problem, we compute $y_{k+1}$ by solving

$$\text{minimize} \quad \left\| \left(I - A(y_k)A(y_k)^+\right) \left(B(x_k, y_k)(y - y_k) - b(y_k)\right) \right\|_2^2$$

from $y_{k+1}$, we then find

$$\begin{aligned} x_{k+1} &= A(y_k)^+ \left(b(y_k) - B(x_k, y_k)(y_{k+1} - y_k)\right) \\ &= \operatorname*{argmin}_x \|A(y_k)x + B(x_k, y_k)(y_{k+1} - y_k) - b(y_k)\|_2^2 \end{aligned}$$

# Variable projection algorithm (VARPRO)

$$\text{minimize} \quad \|f(x,y)\|_2^2 = \|A(y)x - b(y)\|_2^2$$

- we can also eliminate $x$ in the original nonlinear LS problem, before linearizing
- substituting $x = A(y)^+ b(y)$ gives an equivalent nonlinear least squares problem

$$\text{minimize} \quad \left\| \left(I - A(y)A(y)^+\right) b(y) \right\|_2^2$$

- the Gauss–Newton applied to this problem is known as *variable projection*
- to improve convergence, we can add a step size or use Levenberg–Marquardt

# Simplified variable projection

a further simplification results in the following iteration

1. compute $\hat{x} = A(y_k)^+ b(y_k)$, by solving the linear least squares problem

$$\text{minimize} \quad \|A(y_k)x - b(y_k)\|_2^2$$

2. compute $y_{k+1}$ as the solution $y$ of a second linear least squares problem

$$\text{minimize} \quad \left\|\left(I - A(y_k)A(y_k)^+\right)(B(\hat{x}, y_k)(y - y_k) - b(y_k))\right\|_2^2$$

## Interpretation

- step 2 is equivalent to solving the linear least squares problem

$$\text{minimize} \quad \left\|\begin{bmatrix} A(y_k) & B(\hat{x}, y_k) \end{bmatrix} \begin{bmatrix} x \\ y - y_k \end{bmatrix} - b(y_k)\right\|_2^2$$

  in the variables $x$, $y$, and using the solution $y$ as $y_{k+1}$

- *cf.*, GN update of p. 18.18: we replace $x_k$ in $B(x_k, y_k)$ with a better estimate $\hat{x}$

# References


- Å. Björck, *Numerical Methods for Least Squares Problems* (1996), chapter 9.
- J. E. Dennis, Jr., and R. B. Schabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (1996), chapter 10.
- G. Golub and V. Pereyra, *Separable nonlinear least squares: the variable projection method and its applications*, Inverse Problems (2003).
- J. Nocedal and S. J. Wright, *Numerical Optimization* (2006), chapter 10.