18. Gauss–Newton method

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

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

- \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) 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 \( f(x) = Ax - b \)

\[
x^* = A^+ b, \quad g(x^*) = \|(I - AA^+)b\|_2^2 = b^T (I - AA^+)b
\]

\( A^+ \) 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

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

- 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

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

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

- 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 $A \in \mathbb{R}^{2 \times 3}$, $b \in \mathbb{R}^2$, $c \in \mathbb{R}^3$, $d \in \mathbb{R}$

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

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

- 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

$$x_{k+1} = x_k - (f'(x_k)^T f'(x_k))^{-1} f'(x_k)^T f(x_k)$$

$$= x_k - f'(x_k)^+ f(x_k)$$

• 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 $x_{k+1} = x_k + t_kv_k$
Newton and Gauss–Newton steps

\[ g(x) = \| f(x) \|_2^2 = \sum_{i=1}^{m} f_i(x)^2 \]

**Newton step** at \( x = x_k \):

\[
v_{nt} = -\nabla^2 g(x)^{-1} \nabla g(x) = -\left( f'(x)^T f'(x) + \sum_{i=1}^{m} f_i(x) \nabla^2 f_i(x) \right)^{-1} f'(x)^T f(x)
\]

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

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

- this can be written as \( v_{gn} = -H^{-1} \nabla g(x) \) 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: \( H = 2f'(x)^T f'(x) > 0 \) (if \( f'(x) \) has full column rank)
- local convergence to \( x^* \) is similar to Newton method if

\[
\sum_{i=1}^{m} f_i(x^*) \nabla^2 f_i(x^*)
\]

is small (e.g., \( f(x^*) \) is small, or \( f \) is nearly affine around \( x^* \))
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

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

• 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 $k$

\[
\text{minimize } \left\| f'(x_k)(x - x_k) + f(x_k) \right\|^2 + \lambda_k \left\| x - x_k \right\|^2
\]

- 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

\[
v_k = -\left( f'(x_k)^T f(x_k) + \lambda_k I \right)^{-1} f'(x_k)^T f(x_k)
\]

\[
= -\frac{1}{2} \left( f'(x_k)^T f'(x_k) + \lambda_k I \right)^{-1} \nabla g(x_k)
\]

- 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

\[
\text{minimize } \|f'(x_k)v + f(x_k)\|^2_2 + \lambda_k \|v\|^2_2
\]

- 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

\[
\text{minimize } \|f'(x_k)v + f(x_k)\|^2_2 \\
\text{subject to } \|v\|_2 \leq \gamma
\]
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:

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

• $\beta_1, \beta_2$ 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

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

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

Example: the separable structure is common in model fitting problems

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

- 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 \( a_i(y) \in \mathbb{R}^n \):

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

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

\[
f'(x, y) = \begin{bmatrix}
A(y) & B(x, y)
\end{bmatrix}, \quad \text{where } B(x, y) = \begin{bmatrix}
x^T a'_1(y) \\
\vdots \\
x^T a'_m(y)
\end{bmatrix} + b'(y)
\]

here \( a'_i(y) \in \mathbb{R}^{n \times p} \) and \( b'(y) \in \mathbb{R}^{m \times p} \) 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

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

from \( y_{k+1} \), we then find

\[
x_{k+1} = A(y_k)^+ (b(y_k) - B(x_k, y_k)(y_{k+1} - y_k))
\]

\[
= \arg\min_x \| A(y_k)x + B(x_k, y_k)(y_{k+1} - y_k) - b(y_k) \|^2_2
\]
Variable projection algorithm (VARPRO)

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

• 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

\[
\text{minimize } \| (I - A(y)A(y)^+) b(y) \|_2^2
\]

• the Gauss–Newton applied to this problem is known as \textit{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

   $$\min \| A(y_k)x - b(y_k) \|_2^2$$

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

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

Interpretation

• Step 2 is equivalent to solving the linear least squares problem

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

   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