13. Nonlinear least squares

- definition and examples
- derivatives and optimality condition
- Gauss-Newton method
- Levenberg-Marquardt method
Nonlinear least squares

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

- \( f_1(x), \ldots, f_m(x) \) are differentiable functions of a vector variable \( x \)
- \( f \) is a function from \( \mathbb{R}^n \) to \( \mathbb{R}^m \) with components \( f_i(x) \):

\[
f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{bmatrix}
\]

- problem reduces to (linear) least squares if \( f(x) = Ax - b \)
Location from range measurements

• vector $x$ represents unknown location in 2-D or 3-D
• we estimate $x$ by measuring distances to known points $a_1, \ldots, a_m$:

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

• $v_i$ is measurement error

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

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

this is a nonlinear least squares problem with $f_i(x) = \|x - a_i\| - \rho_i$
Example

- Correct position is $(1, 1)$; the five points $a_i$ are marked with blue dots.
- Red square marks (global) minimum at $(1.18, 0.82)$.
**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

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

$c^T x + d > 0$ 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 $x$ 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

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

- $v_i$ is measurement or quantization error
- goal is to determine 3-D location $x$ from the $l$ observations $y_1, \ldots, y_l$

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

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

this is a nonlinear least squares problem with $m = 2l$,

$$f_i(x) = \frac{(A_i x + b_i)_1}{c_i^T x + d_i} - (y_i)_1, \quad f_{l+i}(x) = \frac{(A_i x + b_i)_2}{c_i^T x + d_i} - (y_i)_2$$
Model fitting

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

- model \( \hat{f}(x, \theta) \) is parameterized by parameters \( \theta_1, \ldots, \theta_p \)
- \((x_1, y_1), \ldots, (x_N, y_N)\) are data points
- the minimization is over the model parameters \( \theta \)
- on page 9-9 we considered models that are linear in the parameters \( \theta \):
  \[ \hat{f}(x, \theta) = \theta_1 f_1(x) + \cdots + \theta_p f_p(x) \]
  here we allow \( \hat{f}(x, \theta) \) to be a nonlinear function of \( \theta \)
Example

\[ \hat{f}(x, \theta) = \theta_1 \exp(\theta_2 x) \cos(\theta_3 x + \theta_4) \]

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

\[
\minimize \sum_{i=1}^{N} \left( \theta_1 e^{\theta_2 x_i} \cos(\theta_3 x_i + \theta_4) - y_i \right)^2
\]
Orthogonal distance regression

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

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

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

standard least squares fit  

orthogonal distance fit
Nonlinear least squares formulation

\[
\text{minimize} \quad \sum_{i=1}^{N} \left( (\hat{f}(u_i, \theta) - y_i)^2 + \|u_i - x_i\|^2 \right)
\]

- 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 \((u_i, \hat{f}(u_i, \theta))\)

\[
d_i^2 = (\hat{f}(u_i, \theta) - y_i)^2 + \|u_i - x_i\|^2
\]

- minimizing over \( u_i \) gives squared distance of \((x_i, y_i)\) to graph
- minimizing over \( u \) and \( \theta \) minimizes mean squared distance
Binary classification

\[
\hat{f}(x, \theta) = \text{sign} (\theta_1 f_1(x) + \theta_2 f_2(x) + \cdots + \theta_p f_p(x))
\]

- In lecture 9 (p. 9-25) we computed \(\theta\) by solving a linear least squares problem
- Better results are obtained by solving a nonlinear least squares problem

\[
\text{minimize } \sum_{i=1}^{N} (\phi(\theta_1 f_1(x_i) + \cdots + \theta_p f_p(x_i)) - y_i)^2
\]

- \((x_i, y_i)\) are data points, \(y_i \in \{-1, 1\}\)
- \(\phi(u)\) is the sigmoidal function

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

A differentiable approximation of \(\text{sign}(u)\)
Outline

- definition and examples

- derivatives and optimality condition

- Gauss-Newton method

- Levenberg-Marquardt method
Gradient

Gradient of differentiable function $g : \mathbb{R}^n \to \mathbb{R}$ at $z \in \mathbb{R}^n$ is

$$\nabla g(z) = \left(\frac{\partial g}{\partial x_1}(z), \frac{\partial g}{\partial x_2}(z), \ldots, \frac{\partial g}{\partial x_n}(z)\right)$$

Affine approximation (linearization) of $g$ around $z$ is

$$\hat{g}(x) = g(z) + \frac{\partial g}{\partial x_1}(z)(x_1 - z_1) + \cdots + \frac{\partial g}{\partial x_n}(z)(x_n - z_n)$$

$$= g(z) + \nabla g(z)^T(x - z)$$

(see page 1-30)
Derivative matrix

Derivative matrix (Jacobian) of differentiable function \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) at \( z \in \mathbb{R}^n \):

\[
Df(z) = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1}(z) & \frac{\partial f_1}{\partial x_2}(z) & \ldots & \frac{\partial f_1}{\partial x_n}(z) \\
\frac{\partial f_2}{\partial x_1}(z) & \frac{\partial f_2}{\partial x_2}(z) & \ldots & \frac{\partial f_2}{\partial x_n}(z) \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_m}{\partial x_1}(z) & \frac{\partial f_m}{\partial x_2}(z) & \ldots & \frac{\partial f_m}{\partial x_n}(z)
\end{bmatrix} = \begin{bmatrix}
\nabla f_1(z)^T \\
\nabla f_2(z)^T \\
\vdots \\
\nabla f_m(z)^T
\end{bmatrix}
\]

Affine approximation (linearization) of \( f \) around \( z \) is

\[
\hat{f}(x) = f(z) + Df(z)(x - z)
\]

- see page 3-40
- we also use notation \( \hat{f}(x; z) \) to indicate the point \( z \) around which we linearize
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 \( z \):

\[ \nabla g(z) = \begin{bmatrix} \frac{\partial g}{\partial x_1}(z) \\ \vdots \\ \frac{\partial g}{\partial x_n}(z) \end{bmatrix} = 2 \sum_{i=1}^{m} f_i(z) \nabla f_i(z) = 2Df(z)^Tf(z) \]
Optimality condition

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

- necessary condition for optimality: if \( x \) minimizes \( g(x) \) then it must satisfy

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

- this generalized 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
Outline

- definition and examples
- derivatives and optimality condition
- **Gauss-Newton method**
- Levenberg-Marquardt method
Gauss-Newton method

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

start at some initial guess \(x^{(1)}\), and repeat for \(k = 1, 2, \ldots\):

- linearize \(f\) around \(x^{(k)}\):

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

- substitute affine approximation \(\hat{f}(x; x^{(k)})\) for \(f\) in least squares problem:

\[
\text{minimize } \| \hat{f}(x; x^{(k)}) \|^2
\]

- define \(x^{(k+1)}\) as the solution of this (linear) least squares problem
Gauss-Newton update

least squares problem solved in iteration $k$:

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

- if $Df(x^{(k)})$ has linearly independent columns, solution is given by

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

- Gauss-Newton step $\Delta x^{(k)} = x^{(k+1)} - x^{(k)}$ is

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

$$= - \frac{1}{2} \left( Df(x^{(k)})^T Df(x^{(k)}) \right)^{-1} \nabla g(x^{(k)})$$

(using the expression for $\nabla g(x)$ on page 13-14)
Predicted cost reduction in iteration $k$

• predicted cost function at $x^{(k+1)}$, based on approximation $\hat{f}(x; x^{(k)})$:

$$
\| \hat{f}(x^{(k+1)}; x^{(k)}) \|_2^2 \\
= \| f(x^{(k)}) + Df(x^{(k)}) \Delta x^{(k)} \|_2^2 \\
= \| f(x^{(k)}) \|_2^2 + 2 f(x^{(k)})^T Df(x^{(k)}) \Delta x^{(k)} + \| Df(x^{(k)}) \Delta x^{(k)} \|_2^2 \\
= \| f(x^{(k)}) \|_2^2 - \| Df(x^{(k)}) \Delta x^{(k)} \|_2^2
$$

• if columns of $Df(x^{(k)})$ are linearly independent and $\Delta x^{(k)} \neq 0$,

$$
\| \hat{f}(x^{(k+1)}; x^{(k)}) \|_2^2 < \| f(x^{(k)}) \|_2^2
$$

• however, $\hat{f}(x; x^{(k)})$ is only a local approximation of $f(x)$, so it is possible that

$$
\| f(x^{(k+1)}) \|_2^2 > \| f(x^{(k)}) \|_2^2
$$
Outline

- definition and examples
- derivatives and optimality condition
- Gauss-Newton method
- Levenberg-Marquardt method
Levenberg-Marquardt method

addresses two difficulties in Gauss-Newton method:

- how to update $x^{(k)}$ when columns of $Df(x^{(k)})$ are linearly dependent
- what to do when the Gauss-Newton update does not reduce $\|f(x)\|^2$

Levenberg-Marquardt method

compute $x^{(k+1)}$ by solving a regularized least squares problem

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

- as before, $\hat{f}(x; x^{(k)}) = f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})$
- with $\lambda^{(k)} > 0$, always has a unique solution (no condition on $Df(x^{(k)})$)
- parameter $\lambda^{(k)} > 0$ controls size of update
Levenberg-Marquardt update

regularized least squares problem solved in iteration $k$:

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

- solution is given by

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

- Levenberg-Marquardt step $\Delta x^{(k)} = x^{(k+1)} - x^{(k)}$ is

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

$$= -\frac{1}{2} \left( Df(x^{(k)})^T Df(x^{(k)}) + \lambda^{(k)} I \right)^{-1} \nabla g(x^{(k)})$$

- for $\lambda^{(k)} = 0$ this is the Gauss-Newton step (if defined); for large $\lambda^{(k)}$,

$$\Delta x^{(k)} \approx -\frac{1}{2 \lambda^{(k)}} \nabla g(x^{(k)})$$
Cost reduction

- Predicted cost function at $x^{(k+1)}$, based on local approximation $\hat{f}(x; x^{(k)})$:

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

$$= \|f(x^{(k)}) + Df(x^{(k)})\Delta x^{(k)}\|^2$$

$$= \|f(x^{(k)})\|^2 + 2f(x^{(k)})^T Df(x^{(k)})\Delta x^{(k)} + \|Df(x^{(k)})\Delta x^{(k)}\|^2$$

$$= \|f(x^{(k)})\|^2 - 2\lambda^{(k)}\|\Delta x^{(k)}\|^2 - \|Df(x^{(k)})\Delta x^{(k)}\|^2$$

- For large $\lambda^{(k)}$, we can use affine approximation to estimate actual cost:

$$g(x^{(k+1)}) \approx g(x^{(k)}) + \nabla g(x^{(k)})^T \Delta x^{(k)}$$

$$\|f(x^{(k+1)})\|^2 \approx \|f(x^{(k)})\|^2 + \nabla g(x^{(k)})^T \Delta x^{(k)}$$

$$\approx \|f(x^{(k)})\|^2 - \frac{1}{2\lambda^{(k)}}\|\nabla g(x^{(k)})\|^2$$

(using the expression on page 13-20)
Regularization parameter

several strategies for adapting \( \lambda^{(k)} \) are possible; the simplest example:

- at iteration \( k \), compute the solution \( \hat{x} \) of
  \[
  \text{minimize} \quad \| f(x; x^{(k)}) \|^2 + \lambda^{(k)} \| x - x^{(k)} \|^2
  \]

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

Some variations

- compare actual cost reduction with predicted cost reduction
- solve a least squares problem with ‘trust region’
  \[
  \text{minimize} \quad \| f(x; x^{(k)}) \|^2 \\
  \text{subject to} \quad \| x - x^{(k)} \|^2 \leq \gamma
  \]
Summary: Levenberg-Marquardt method

choose $x^{(1)}$ and $\lambda^{(1)}$ and repeat for $k = 1, 2, \ldots$:

1. evaluate $f(x^{(k)})$ and $A = Df(x^{(k)})$

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 $0 < \beta_1 < 1 < \beta_2$

- in step 2, $\hat{x}$ can be computed using a QR factorization

- terminate if $\nabla g(x^{(k)}) = 2A^T f(x^{(k)})$ is sufficiently small
Location from range measurements

- iterates from three starting points, with $\lambda^{(1)} = 0.1$, $\beta_1 = 0.8$, $\beta_2 = 2$
- algorithm started at $(1.8, 3.5)$ and $(3.0, 1.5)$ finds minimum $(1.18, 0.82)$
- started at $(2.2, 3.5)$ converges to non-optimal point
Cost function and regularization parameter

cost function and $\lambda^{(k)}$ for the three starting points on previous page