7. Least squares

- linear least-squares
- regularized least-squares
- nonlinear least squares
- Gauss-Newton method
- Levenberg-Marquardt method

Linear least-squares

Inconsistent linear equations

$$A\boldsymbol{x} = \boldsymbol{b}$$

- $A = [a_{ij}] \in \mathbb{R}^{m \times n}$ is tall matrix $m > n$ and $\boldsymbol{b} = (b_1, \ldots, b_m) \in \mathbb{R}^m$
- if the system is *inconsistent* ($\operatorname{rank} A \neq \operatorname{rank}[A\ \boldsymbol{b}]$), then it has no solution and it is desirable to find an $\boldsymbol{x}$ such that $A\boldsymbol{x} \approx \boldsymbol{b}$

(linear) **Least squares problem**

$$\text{minimize} \quad \|A\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} x_j - b_i \right)^2 \tag{7.1}$$

- $\boldsymbol{r} = A\boldsymbol{x} - \boldsymbol{b}$ is called the *residual*
- A and $\boldsymbol{b}$ are normally called the *data* for the problem

Column and row interpretations

let $\boldsymbol{a}_i$ denote the ith column of A and $\hat{\boldsymbol{a}}_j^T$ denote the jth row of A:

$$A = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_n \end{bmatrix} \quad \text{or} \quad A = \begin{bmatrix} \hat{\boldsymbol{a}}_1^T \\ \vdots \\ \hat{\boldsymbol{a}}_m^T \end{bmatrix}$$

Row interpretation

$$\text{minimize} \quad \|A\boldsymbol{x} - \boldsymbol{b}\|^2 = (\hat{\boldsymbol{a}}_1^T\boldsymbol{x} - b_1)^2 + \cdots + (\hat{\boldsymbol{a}}_m^T\boldsymbol{x} - b_m)^2$$

minimize the sum of squares of the residuals $r_i = \hat{\boldsymbol{a}}_i^T\boldsymbol{x} - b_i$

Column interpretation

$$\text{minimize} \quad \|A\boldsymbol{x} - \boldsymbol{b}\|^2 = \|(x_1\boldsymbol{a}_1 + \cdots + x_n\boldsymbol{a}_n) - \boldsymbol{b}\|^2$$

find the coefficients of the linear combination of the columns that is closest to the m vector $\boldsymbol{b}$

Solution

Normal equations: the solution of the least squares problem must satisfy the *normal equations*

$$A^T A \boldsymbol{x}^\star = A^T \boldsymbol{b} \qquad (7.2)$$

- any $\boldsymbol{x}$ satisfying (7.2) is a global minimizer since $\nabla^2 f(\boldsymbol{x}) = 2A^T A \geq 0$
- if the columns of A are linearly independent, then the solution is unique:

$$\boldsymbol{x}^\star = (A^T A)^{-1} A^T \boldsymbol{b}$$

MATLAB command

```
>> A=[] % define the matrix A
>> b=[] % define the vector b
>> x=A\b % solution
```

Example 7.1

we are given two different types of concrete:

- the first type contains 30% cement, 40% gravel, and 30% sand (all percentages of weight)
- the second type contains 10% cement, 20% gravel, and 70% sand

how many pounds of each type of concrete should you mix together so that you get a concrete mixture that has as close as possible to a total of 5 pounds of cement, 3 pounds of gravel, and 4 pounds of sand?

- letting x_1 and x_2 to be the amounts of concrete of the first and second types, the above problem can be formulated as the least squares problem:

$$\text{minimize} \quad \left\| \begin{bmatrix} 0.3 & 0.1 \\ 0.4 & 0.2 \\ 0.3 & 0.7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 5 \\ 3 \\ 4 \end{bmatrix} \right\|^2 = \|A\boldsymbol{x} - \boldsymbol{b}\|^2,$$

  where $\boldsymbol{x} = (x_1, x_2)$

- since the columns of A are linearly independent, the solution is

$$\boldsymbol{x}^\star = (A^T A)^{-1} A^T \boldsymbol{b} = \begin{bmatrix} 10.6 \\ 0.961 \end{bmatrix}$$

Optimality verification using algebra

$$
\begin{aligned}
\|A\boldsymbol{x}-\boldsymbol{b}\|^2 &= \|(A\boldsymbol{x}-A\boldsymbol{x}^\star)+(A\boldsymbol{x}^\star-\boldsymbol{b})\|^2 \\
&= \|A(\boldsymbol{x}-\boldsymbol{x}^\star)\|^2+\|A\boldsymbol{x}^\star-\boldsymbol{b}\|^2 \\
&\quad + 2(A\boldsymbol{x}-A\boldsymbol{x}^\star)^T(A\boldsymbol{x}^\star-\boldsymbol{b})
\end{aligned}
$$

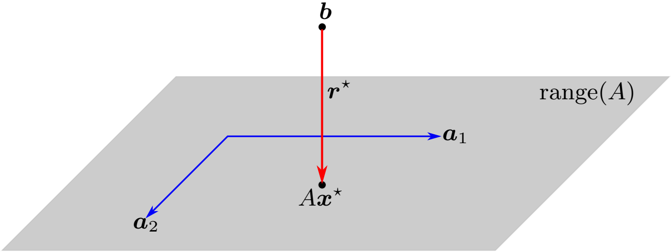
using $A^TA\boldsymbol{x}^\star = A^T\boldsymbol{b}$, the cross product term is zero; this implies that

$$
\|A\boldsymbol{x}-\boldsymbol{b}\|^2 = \|A(\boldsymbol{x}-\boldsymbol{x}^\star)\|^2+\|A\boldsymbol{x}^\star-\boldsymbol{b}\|^2
$$

- since $\|A(\boldsymbol{x}-\boldsymbol{x}^\star)\|^2 \geq 0$, we have $\|A\boldsymbol{x}-\boldsymbol{b}\|^2 \geq \|A\boldsymbol{x}^\star-\boldsymbol{b}\|^2$
- if the columns of A are linearly independent, then $\|A(\boldsymbol{x}-\boldsymbol{x}^\star)\|^2 > 0$ and $\|A\boldsymbol{x}-\boldsymbol{b}\|^2 > \|A\boldsymbol{x}^\star-\boldsymbol{b}\|^2$ for $\boldsymbol{x} \neq \boldsymbol{x}^\star$

Geometric interpretation

Orthogonality principle: the optimal residual $\boldsymbol{r}^\star = A\boldsymbol{x}^\star - \boldsymbol{b}$ is orthogonal to the columns of A



for any n-vector $\boldsymbol{v}$, then we have

$$(A\boldsymbol{v})^T\boldsymbol{r}^\star = (A\boldsymbol{v})^T(A\boldsymbol{x}^\star - \boldsymbol{b}) = \boldsymbol{v}^T A^T(A\boldsymbol{x}^\star - \boldsymbol{b}) = \boldsymbol{v}^T\boldsymbol{0} = \boldsymbol{0},$$

where the zero is due to the normal equation (7.2)

Data fitting

given m data points $(\boldsymbol{z}_i, y_i)$ where $\boldsymbol{z}_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$, we want to find a function $g : \mathbb{R}^n \to \mathbb{R}$ such that

$$g(\boldsymbol{z}_i) \approx y_i, \quad i = 1, \dots, m \tag{7.3}$$

assume that the function g has the linear structure

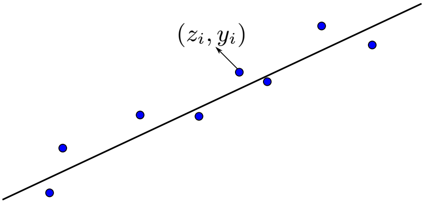
$$g(\boldsymbol{z}) = x_1 g_1(\boldsymbol{z}) + x_2 g_2(\boldsymbol{z}) + \cdots + x_n g_n(\boldsymbol{z})$$

- $g_i(\boldsymbol{z})$ are given functions, referred to as *basis functions*
- x_i are unknown parameters
- we want to estimate $\boldsymbol{x}$ such that the approximation (7.3) is "good"

Least-squares formulation: minimize $\|A\boldsymbol{x} - \boldsymbol{b}\|^2$ where

$$A = \begin{bmatrix} g_1(\boldsymbol{z}_1) & g_2(\boldsymbol{z}_1) & \cdots & g_n(\boldsymbol{z}_1) \\ g_1(\boldsymbol{z}_2) & g_2(\boldsymbol{z}_2) & \cdots & g_n(\boldsymbol{z}_2) \\ \vdots & \vdots & & \vdots \\ g_1(\boldsymbol{z}_m) & g_2(\boldsymbol{z}_m) & \cdots & g_n(\boldsymbol{z}_m) \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Line fitting



find a straight line that best fits the data (z_i, y_i):

$$x_1 + x_2 z_i \approx y_i$$

- x_1 is the displacement
- x_2 is the slope of the line
- $g(z) = x_1 + x_2 z$, $g_1(z) = 1$, $g_2(z) = z$

$$A = \begin{bmatrix} 1 & z_1 \\ 1 & z_2 \\ \vdots & \vdots \\ 1 & z_m \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Example 7.2

we want fit a straight line $y_i \approx x_1 + x_2 z_i$ to the data:

$$(z_1, y_1) = (2, 3), \quad (z_2, y_2) = (3, 4), \quad (z_3, y_3) = (4, 15)$$

- we can minimize

$$\sum_{i=1}^{3} (x_1 + x_2 z_i - y_i)^2$$
$$= (x_1 + 2x_2 - 3)^2 + (x_1 + 3x_2 - 4)^2 + (x_1 + 4x_2 - 15)^2 = \|A\boldsymbol{x} - \boldsymbol{b}\|^2$$

where

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 3 \\ 4 \\ 15 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- the solution is

$$\boldsymbol{x}^\star = \begin{bmatrix} x_1^\star \\ x_2^\star \end{bmatrix} = (A^T A)^{-1} A^T \boldsymbol{b} = \begin{bmatrix} -32/3 \\ 6 \end{bmatrix}$$

Linear estimation (regression)

we have m measurements $y_1, \ldots, y_m$ of some time-varying linear system:

$$y_t = \boldsymbol{h}_t^T \boldsymbol{x} + v_t, \quad t = 1, \ldots, m$$

where $\boldsymbol{h}_t^T$ are known or measured linear system parameters, and v_t is an unknown small measurement noise

- the estimation problem is to find a good $\boldsymbol{x}$ such that $y_t - \boldsymbol{h}_t^T \boldsymbol{x}$ is minimized for all t
- we can formulate this as a least square problem with

$$A = \begin{bmatrix} \boldsymbol{h}_1^T \\ \vdots \\ \boldsymbol{h}_m^T \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Example 7.3

- we apply a 1-ampere current through the resistor and measure a noisy voltage across it
- we have n measurements

$$V_i = R + n_i \qquad i = 1, \ldots, n$$

we wish to find R that best fits our measurements

this problem can be formulated as

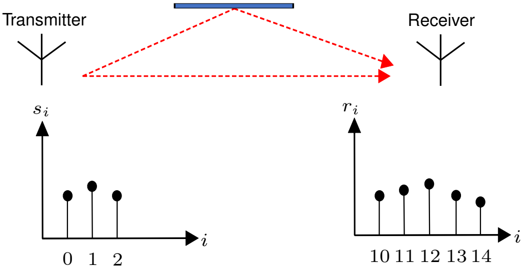
$$\text{minimize} \quad \left\| \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} R - \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \right\|^2$$

least-squares problem with $A = \mathbf{1}$ and $\boldsymbol{b} = (V_1, \ldots, V_n)$; hence solution is

$$R^\star = (A^T A)^{-1} A^T \boldsymbol{b} = \frac{1}{n} \sum_{i=1}^{n} V_i$$

Example 7.4

- a wireless transmitter sends three signals s_0, s_1, and s_2 at times $t = 0, 1, 2$; the transmitted signal takes two paths to the receiver:
  I. direct path, with delay 10 and attenuation factor α_1
  II. indirect (reflected) path, with delay 12 and attenuation factor α_2
- the received signal is measured from times $t = 10$ to $t = 14$, which is the sum of the signals from these two paths, with their respective delays and attenuation factors plus some unknown noise



find the channel attenuation factors α_1 and α_2 that “best” fits the signals:

$$\boldsymbol{s} = (s_0, s_1, s_2) = (1, 2, 1)$$
$$(r_{10}, r_{11}, r_{12}, r_{13}, r_{14}) = (4, 7, 8, 6, 3)$$

we can formulate this as a least-squares problem with

$$A = \begin{bmatrix} s_0 & 0 \\ s_1 & 0 \\ s_2 & s_0 \\ 0 & s_1 \\ 0 & s_2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} r_{10} \\ r_{11} \\ r_{12} \\ r_{13} \\ r_{14} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}$$

the least-squares solution is

$$\begin{aligned} \boldsymbol{x}^\star &= (A^T A)^{-1} A^T \boldsymbol{b} \\ &= \begin{bmatrix} \|\boldsymbol{s}\|^2 & s_0 s_2 \\ s_0 s_2 & \|\boldsymbol{s}\|^2 \end{bmatrix}^{-1} \begin{bmatrix} s_0 r_{10} + s_1 r_{11} + s_0 r_{12} \\ s_0 r_{12} + s_1 r_{13} + s_0 r_{14} \end{bmatrix} \\ &= \begin{bmatrix} 6 & 1 \\ 1 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 4 + 14 + 8 \\ 8 + 12 + 3 \end{bmatrix} = \begin{bmatrix} \frac{133}{35} \\ \frac{112}{35} \end{bmatrix} \end{aligned}$$

Outline

- linear least-squares
- **regularized least-squares**
- nonlinear least squares
- Gauss-Newton method
- Levenberg-Marquardt method

Regularized least-squares

$$\text{minimize} \quad \|A\boldsymbol{x} - \boldsymbol{b}\|^2 + \rho\|R\boldsymbol{x}\|^2$$

- $R \in \mathbb{R}^{p \times n}$ is the *regularization matrix* and ρ is the *regularization parameter*
- large ρ gives more emphasis on making the term $\rho\|R\boldsymbol{x}\|^2$ small

Why regularization?

- utilize some prior information about $\boldsymbol{x}$
- useful for algorithm implementations

Solution:

$$(A^T A + \rho R^T R)\boldsymbol{x} = A^T \boldsymbol{b}$$

if $A^T A + \rho R^T R$ is invertible, then

$$\boldsymbol{x}^\star = (A^T A + \rho R^T R)^{-1} A^T \boldsymbol{b}$$

Example: signal de-noising

- $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ represent some signal (*e.g.*, audio signals)
- x_i represents the value of the signal sampled at time i
- the signal can be measured with some additive noise

$$\boldsymbol{s} = \boldsymbol{x} + \boldsymbol{v}$$

  where $\boldsymbol{v}$ is some noise
- the signal does not vary too much $|x_{i+1} - x_i| << 1$
- given $\boldsymbol{s}$, we want to find a "good" estimate of $\boldsymbol{x}$

Naive solution: directly set $\boldsymbol{x} = \boldsymbol{s}$; however, this can result in a bad estimate if some noise components v_i are large

Least-squares formulation

$$\text{minimize} \quad \|\boldsymbol{x} - \boldsymbol{s}\|^2 + \rho\|R\boldsymbol{x}\|^2$$

- ρ is a smoothing regularization parameter
- R is an $(n-1) \times n$ smoothing matrix:

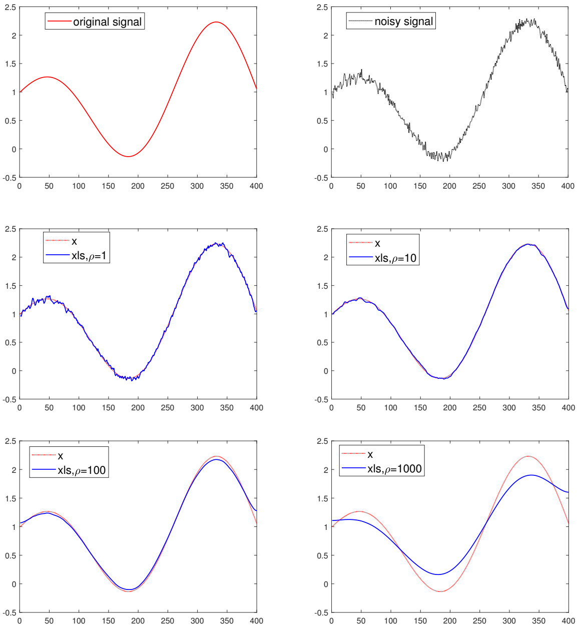
$$\|R\boldsymbol{x}\|^2 = \sum_{i=1}^{n-1}(x_i - x_{i+1})^2$$

the matrix R has the structure

$$R = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \in \mathbb{R}^{(n-1)\times n}$$

- the optimal solution is given by

$$\boldsymbol{x}^\star(\rho) = (I + \rho R^T R)^{-1}\boldsymbol{s}$$


original signal
noisy signal
x
xls,ρ=1
x
xls,ρ=10
x
xls,ρ=100
x
xls,ρ=1000

Outline

- linear least-squares
- regularized least-squares
- **nonlinear least squares**
- Gauss-Newton method
- Levenberg-Marquardt method

Nonlinear least squares

$$\text{minimize} \quad \|r(\boldsymbol{x})\|^2 = r_1(\boldsymbol{x})^2 + \cdots + r_m(\boldsymbol{x})^2$$

- $r : \mathbb{R}^n \to \mathbb{R}^m$ is nonlinear function with components $r_i : \mathbb{R}^n \to \mathbb{R}$
- when $r(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b}$, we recover the linear least-squares problem
- nonlinear least squares are hard to solve
- solution solves/approximate the solution to a set of m *nonlinear* equations:

$$r_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, m$$

Location from distance of measurements

- locate some object with unknown location $\boldsymbol{x} \in \mathbb{R}^n$ ($n = 2$ or $n = 3$)
- we have some noisy measurements of the distance to from $\boldsymbol{x}$ to some known locations $\boldsymbol{y}_i$:

$$\gamma_i = \|\boldsymbol{x} - \boldsymbol{y}_i\| + v_i, \quad i = 1, \ldots, m$$

  where v_i is some small measurement noise
- we can estimate the position of $\boldsymbol{x}$ by solving

$$\text{minimize} \quad \sum_{i=1}^{m} (\|\boldsymbol{x} - \boldsymbol{y}_i\| - \gamma_i)^2$$

  this is a nonlinear least-squares problem with $r_i(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{y}_i\| - \gamma_i$

Nonlinear data-fitting

Model fitting problem

- we have m *data points* or *measurements* $(\boldsymbol{z}_i, y_i)$, $i = 1, \ldots, m$, where $\boldsymbol{z}_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}$
- these points are approximately related by the equation

$$g(\boldsymbol{z}_i; \boldsymbol{x}) \approx y_i, \quad i = 1, \ldots, m \tag{7.4}$$

  where $g : \mathbb{R}^n \to \mathbb{R}$ is known and $\boldsymbol{x}$ are unknown parameters

Nonlinear least squares formulation

$$\text{minimize} \quad \sum_{i=1}^{m} (g(\boldsymbol{z}_i; \boldsymbol{x}) - y_i)^2$$

if g is linear in parameters x_i, then we get a linear least-squares

Example 7.5

- given m measurements, $y_1, y_2, \ldots, y_m$, at m points of time, $t_1, \ldots, t_m$ of a sinusoidal signal:

$$y_i = \beta \sin(\omega t_i + \phi) + n(t_i)$$

  where $n(t_i)$ is a random noise

- find the parameters β, ω and ϕ that gives some optimal fit to these measurements

Nonlinear least-squares formulation

$$\text{minimize} \quad \sum_{i=1}^{m} r_i(\boldsymbol{x})^2 = \sum_{i=1}^{m} \big(y_i - \beta \sin(\omega t_i + \phi)\big)^2$$

with variable $\boldsymbol{x} = (\beta, \omega, \phi)$ and $r_i(\boldsymbol{x}) = y_i - \beta \sin(\omega t_i + \phi)$

Classification

Classification problem

- we have m training data points $(\boldsymbol{z}_i, y_i)$, $i = 1, \dots, m$, where y_i can take certain *discrete values*
- we want to fit the data to the model $g(\boldsymbol{z}_i) \approx y_i$
- determine which class the a new data point $\boldsymbol{z}$ belongs to

Boolean classification

- $y \in \{+1, -1\}$
- values of y can represent two categories such as true/false, spam/not spam, dog/cat...etc
- the model $g(\boldsymbol{z}) \approx \boldsymbol{y}$ is called a *Boolean classifier*

Least squares classifier

we are given the data points $(\boldsymbol{z}_i, y_i)$, $i = 1, \dots, m$ and a linear in parameter model

$$g(\boldsymbol{z}) = x_1 g_1(\boldsymbol{z}) + x_2 g_2(\boldsymbol{z}) + \cdots + x_n g_n(\boldsymbol{z})$$

we want to determine whether new data $\boldsymbol{z}_{m+1}$ belong to class $+1$ or class -1

Least squares Boolean classifier

- solve linear least-squares data-fitting problem to find the parameters $x_1, \dots, x_n$
- take the sign of $g(\boldsymbol{z})$ to get the *Boolean classifier:*

$$\hat{g}(\boldsymbol{z}) = \operatorname{sign}(g(\boldsymbol{z})) = \begin{cases} +1 & \text{if } g(\boldsymbol{z}) \geq 0 \\ -1 & \text{if } g(\boldsymbol{z}) < 0 \end{cases}$$

better results if we solve a nonlinear least squares problem

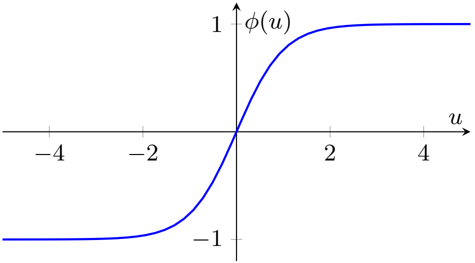
Nonlinear formulation

$$\text{minimize} \quad \sum_{i=1}^{m} \left(\phi\big(x_1 g_1(\boldsymbol{z}_i) + x_2 g_2(\boldsymbol{z}_i) + \cdots + x_n g_n(\boldsymbol{z}_i)\big) - y_i \right)^2$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is the sigmoidal function:

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

which is a differentiable approximation of $\mathrm{sign}(u)$



Outline

- linear least-squares
- regularized least-squares
- nonlinear least squares
- **Gauss-Newton method**
- Levenberg-Marquardt method

Linear least square approximation at each iteration

given an estimate of a solution $\boldsymbol{x}^{(k)}$ at time k, the Gauss-Newton method produces a new estimate $\boldsymbol{x}^{(k+1)}$ that solves the problem

$$\text{minimize} \quad \|\hat{r}(\boldsymbol{x};\boldsymbol{x}^{(k)})\|^2 = \|r(\boldsymbol{x}^{(k)}) + Dr(\boldsymbol{x}^{(k)})(\boldsymbol{x}-\boldsymbol{x}^{(k)})\|^2$$

- $\hat{r}(\boldsymbol{x};\boldsymbol{x}^{(k)})$ is first order Taylor approximation around $\boldsymbol{z}$:

$$r(\boldsymbol{x}) \approx \hat{r}(\boldsymbol{x};\boldsymbol{z}) = r(\boldsymbol{z}) + Dr(\boldsymbol{z})(\boldsymbol{x}-\boldsymbol{z}) \quad \text{if } \boldsymbol{x} \text{ is close to } \boldsymbol{z}$$

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

$$A = Dr(\boldsymbol{x}^{(k)}), \quad \boldsymbol{b} = Dr(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)} - r(\boldsymbol{x}^{(k)})$$

Gauss-Newton method

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

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= (A^T A)^{-1} A^T \boldsymbol{b} \\
&= \left(Dr(\boldsymbol{x}^{(k)})^T Dr(\boldsymbol{x}^{(k)}) \right)^{-1} Dr(\boldsymbol{x}^{(k)})^T \big(Dr(\boldsymbol{x}^{(k)}) \boldsymbol{x}^{(k)} - r(\boldsymbol{x}^{(k)}) \big) \\
&= \boldsymbol{x}^{(k)} - \left(Dr(\boldsymbol{x}^{(k)})^T Dr(\boldsymbol{x}^{(k)}) \right)^{-1} Dr(\boldsymbol{x}^{(k)})^T r(\boldsymbol{x}^{(k)})
\end{aligned}$$

- assumes that $A = Dr(\boldsymbol{x}^{(k)})$ has linearly independent columns
- if converged $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}$, then

$$Dr(\boldsymbol{x}^{(k)})^T r(\boldsymbol{x}^{(k)}) = \boldsymbol{0}$$

  hence $\boldsymbol{x}^{(k)}$ satisfies the optimality condition since the gradient of $\|r(\boldsymbol{x})\|^2$ is $2Dr(\boldsymbol{x})^T r(\boldsymbol{x})$

Stopping criteria

- if $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)}$, then $\boldsymbol{x}^{(k)}$ satisfies the optimality condition
- this does not mean that $\boldsymbol{x}^{(k)}$ is a good solution since it can be a local minimizer, local maximizer, or a saddle-point
- in practice, the algorithm can be stopped if $\|r(\boldsymbol{x}^{(k)})\|^2$ is small enough
- it is also common to run the algorithm from different starting points and choose the best solution of these multiple runs

Gauss-Newton algorithm

Algorithm Gauss-Newton algorithm

given a starting point $\boldsymbol{x}^{(0)}$ and solution tolerance ϵ

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

1. evaluate $Dr(\boldsymbol{x}^{(k)}) = (\nabla r_1(\boldsymbol{x}^{(k)})^T, \ldots, \nabla r_m(\boldsymbol{x}^{(k)})^T)$
2. set
$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left(Dr(\boldsymbol{x}^{(k)})^T Dr(\boldsymbol{x}^{(k)})\right)^{-1} Dr(\boldsymbol{x}^{(k)})^T r(\boldsymbol{x}^{(k)})$$

   if $\|r(\boldsymbol{x}^{(k)})\|^2 \leq \epsilon$ stop and output $\boldsymbol{x}^{(k+1)}$

Gauss-Newton step is

$$\boldsymbol{d}_{\mathrm{gn}} = -\left(Dr(\boldsymbol{x}^{(k)})^T Dr(\boldsymbol{x}^{(k)})\right)^{-1} Dr(\boldsymbol{x}^{(k)})^T r(\boldsymbol{x}^{(k)})$$

Relation to Newton's method

$$f(\boldsymbol{x}) = \tfrac{1}{2}\|r(\boldsymbol{x})\|^2 = \tfrac{1}{2}\left(r_1(\boldsymbol{x})^2 + \cdots + r_m(\boldsymbol{x})^2\right)$$

- gradient and Hessian of the above function are

$$\nabla f(\boldsymbol{z}) = Dr(\boldsymbol{z})^T r(\boldsymbol{z})$$

$$\nabla^2 f(\boldsymbol{z}) = Dr(\boldsymbol{z})^T Dr(\boldsymbol{z}) + \sum_{j=1}^{m} r_j(\boldsymbol{z}) \nabla^2 r_j(\boldsymbol{z})$$

- suppose we approximate the Hessian by

$$\nabla^2 f(\boldsymbol{z}) \approx Dr(\boldsymbol{z})^T Dr(\boldsymbol{z})$$

- then, using this approximation, the (undamped) Newton update becomes

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left(Dr(\boldsymbol{x}^{(k)})^T Dr(\boldsymbol{x}^{(k)})\right)^{-1} Dr(\boldsymbol{x}^{(k)})^T r(\boldsymbol{x}^{(k)})$$

the above update is the basic Gauss-Newton update

Issues with Gauss-Newton method

an advantage of Gauss-Newton is that it only computes first-order derivatives where Newton's method computes the Hessian; however, it has some issues:

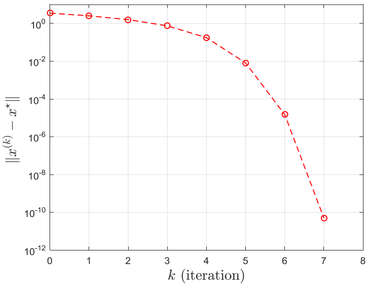
- when $\boldsymbol{x}^{(k+1)}$ is not close to $\boldsymbol{x}^{(k)}$, the affine approximation will not be accurate and the algorithm may fail
- a second major issue is that columns of the matrix $Dr(\boldsymbol{x}^{(k)})$ may not always be linearly independent; in this case, the next iterate is not defined

Numerical Example II

$$r(x) = e^{x} - e^{-x} - 1$$

since $r'(x) = e^{x} + e^{-x}$, the Gauss-Newton iteration is

$$x^{(k+1)} = x^{(k)} - \frac{e^{x^{(k)}} - e^{-x^{(k)}} - 1}{e^{x^{(k)}} + e^{-x^{(k)}}}$$



evolution of the error with initial point at $x^{(0)} = 5$; the algorithm quickly converges to $x^{\star} = 0.4812$

Numerical Example III

$$r_i(\boldsymbol{x}) = \sqrt{(x_1 - p_i)^2 + (x_2 - q_i)^2} - \gamma_i, \quad i = 1, \ldots, 5$$

where p_i, q_i, γ_i are given

the gradient of r_i is

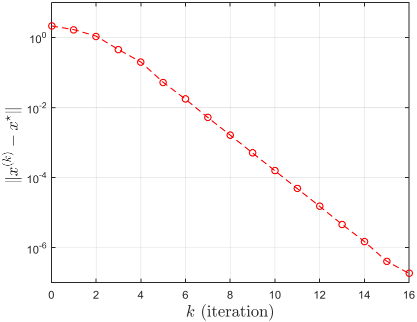
$$\nabla r_i(\boldsymbol{x}) = \begin{bmatrix} \frac{x_1 - p_i}{\sqrt{(x_1-p_i)^2+(x_2-q_i)^2}} \\ \frac{x_2 - q_i}{\sqrt{(x_1-p_i)^2+(x_2-q_i)^2}} \end{bmatrix}$$

thus, the Jacobian of r is

$$Dr(\boldsymbol{x}) = \begin{bmatrix} \frac{x_1-p_1}{\sqrt{(x_1-p_1)^2+(x_2-q_1)^2}} & \frac{x_2-q_1}{\sqrt{(x_1-p_1)^2+(x_2-q_1)^2}} \\ \frac{x_1-p_2}{\sqrt{(x_1-p_2)^2+(x_2-q_2)^2}} & \frac{x_2-q_2}{\sqrt{(x_1-p_2)^2+(x_2-q_2)^2}} \\ \frac{x_1-p_3}{\sqrt{(x_1-p_3)^2+(x_2-q_3)^2}} & \frac{x_2-q_3}{\sqrt{(x_1-p_3)^2+(x_2-q_3)^2}} \\ \frac{x_1-p_4}{\sqrt{(x_1-p_4)^2+(x_2-q_4)^2}} & \frac{x_2-q_4}{\sqrt{(x_1-p_4)^2+(x_2-q_4)^2}} \\ \frac{x_1-p_5}{\sqrt{(x_1-p_5)^2+(x_2-q_5)^2}} & \frac{x_2-q_5}{\sqrt{(x_1-p_5)^2+(x_2-q_5)^2}} \end{bmatrix}$$

where we assume $(x_1, x_2) \neq (p_i, q_i)$

results with data $\boldsymbol{p} = \begin{bmatrix} 8 \\ 2.0 \\ 1.5 \\ 1.5 \\ 2.5 \end{bmatrix}, \quad \boldsymbol{q} = \begin{bmatrix} 5 \\ 1.7 \\ 1.5 \\ 2.0 \\ 1.5 \end{bmatrix}, \quad \boldsymbol{\gamma} = \begin{bmatrix} 1.87 \\ 1.24 \\ 0.53 \\ 1.29 \\ 1.49 \end{bmatrix}$



the evolution of the error with initial point at $x^{(0)} = (1, 3)$; the algorithm converges to solution $\boldsymbol{x}^\star = (1.1833, 0.8275)$

Outline

- linear least-squares
- regularized least-squares
- nonlinear least squares
- Gauss-Newton method
- **Levenberg-Marquardt method**

Regularized approximate problem

$$\text{minimize} \quad \|r(\boldsymbol{x}^{(k)}) + Dr(\boldsymbol{x}^{(k)})(\boldsymbol{x} - \boldsymbol{x}^{(k)})\|^2 + \rho_k \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2$$

- regularization fixes invertibility issue of Gauss-Newton
- regularization parameter ρ_k controls how close $x^{(k+1)}$ is to $x^{(k)}$
- the above problem can be rewritten as

$$\text{minimize} \quad \left\| \begin{bmatrix} Dr(\boldsymbol{x}^{(k)}) \\ \sqrt{\rho_k} I \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} Dr(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)} - r(\boldsymbol{x}^{(k)}) \\ \sqrt{\rho_k}\boldsymbol{x}^{(k)} \end{bmatrix} \right\|^2$$

this is just a least-squares problem with cost $\|A\boldsymbol{x} - \boldsymbol{b}\|^2$ where

$$A = \begin{bmatrix} Dr(\boldsymbol{x}^{(k)}) \\ \sqrt{\rho_k} I \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} Dr(\boldsymbol{x}^{(k)})\boldsymbol{x}^{(k)} - r(\boldsymbol{x}^{(k)}) \\ \sqrt{\rho_k}\boldsymbol{x}^{(k)} \end{bmatrix}$$

the solution is

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left(Dr(\boldsymbol{x}^{(k)})^T Dr(\boldsymbol{x}^{(k)}) + \rho_k I\right)^{-1} Dr(\boldsymbol{x}^{(k)})^T r(\boldsymbol{x}^{(k)})$$

Updating ρ

- if ρ_k is very small, then $x^{(k+1)}$ can be far from $x^{(k)}$, and the method may fail
- if ρ_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 ρ_k is to check whether

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

  if so, then we can decrease ρ_{k+1}; otherwise, we increase ρ_{k+1}

Algorithm Levenberg-Marquardt algorithm

given a starting point $\boldsymbol{x}^{(0)}$, solution tolerance ϵ, and $\rho_0 > 0$

repeat for $k \geq 0$

1. evaluate $Dr(\boldsymbol{x}^{(k)}) = (\nabla r_1(\boldsymbol{x}^{(k)})^T, \ldots, \nabla r_m(\boldsymbol{x}^{(k)})^T)$
2. update

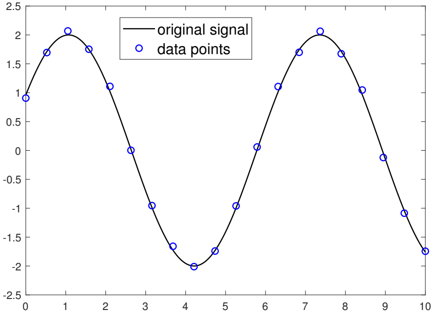
$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left(Dr(\boldsymbol{x}^{(k)})^T Dr(\boldsymbol{x}^{(k)}) + \rho_k I \right)^{-1} Dr(\boldsymbol{x}^{(k)})^T r(\boldsymbol{x}^{(k)})$$

   if $\|r(\boldsymbol{x}^{(k)})\|^2 \leq \epsilon$ stop and output $\boldsymbol{x}^{(k+1)}$

3. if $\|r(\boldsymbol{x}^{(k+1)})\|^2 < \|r(\boldsymbol{x}^{(k)})\|^2$, then decrease ρ_{k+1} (*e.g.*, $\rho_{k+1} = 0.9\rho_k$); otherwise, increase ρ_{k+1} (*e.g.*, $\rho_{k+1} = 10\rho_k$)

Numerical example IV

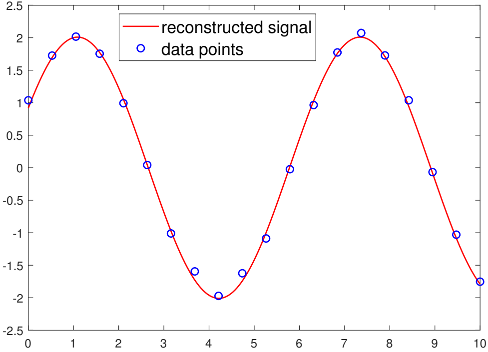
- data-fitting problem with $r_i(\beta, \omega, \phi) = y_i - \beta \sin(\omega t_i + \phi)$
- find (β, ω, ϕ) given $m = 20$ data points



- for this problem, we have

$$\nabla r_i(\beta, \omega, \phi) = \begin{bmatrix} -\sin(\omega t_i + \phi) \\ -\beta t_i \cos(\omega t_i + \phi) \\ -\beta \cos(\omega t_i + \phi) \end{bmatrix}$$

- applying Levenberg-Marquardt algorithm gives



References and further readings

- Stephen Boyd and Lieven Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares,* Cambridge University Press, 2018, chapters 12, 18.
- Edwin KP Chong and Stanislaw H Zak. *An Introduction to Optimization,* John Wiley & Sons, 2013, chapter 12.1.
- Amir Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB,* SIAM, 2014, chapter 3.