7. Least squares

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

Linear least squares

Inconsistent linear equations

- let A be $m \times n$ and consider $Ax = b$ where b is an m-vector
- in most applications, $m > n$ and there is no x satisfying $Ax = b$ ($b \notin \mathrm{range}(A)$)
- it is desirable to find an x such that $Ax \approx b$

(linear) **Least squares:** choose x that solves:

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

- $r = Ax - b$ is called the *residual*
- A and b are called the *data* for the problem
- also called *regression* (in data-fitting context)

Column and row interpretations

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

$$A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} = \begin{bmatrix} \hat{a}_1^T \\ \vdots \\ \hat{a}_m^T \end{bmatrix}$$

Row interpretation

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

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

Column interpretation

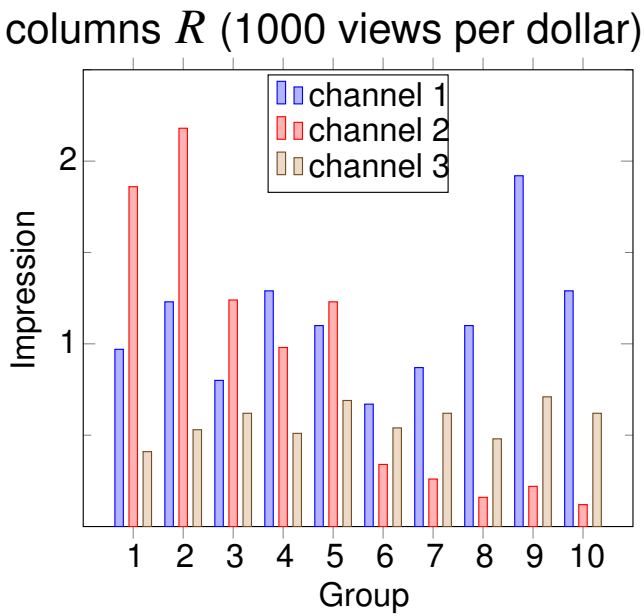
$$\text{minimize} \quad \|Ax - b\|^2 = \|(x_1 a_1 + \cdots + x_n a_n) - b\|^2$$

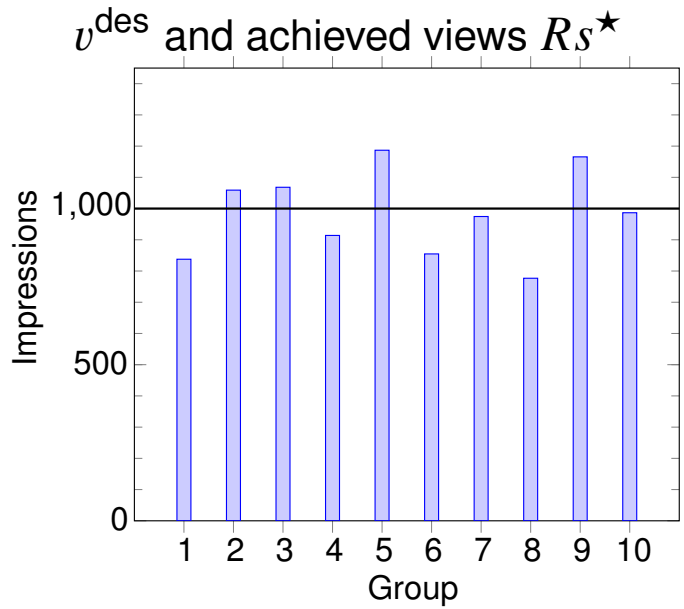
find the coefficients of the linear combination of the columns that is closest to b

Example: advertising purchases

- m demographics groups (audiences), n advertising channels
- v_i^{des} is target number of views or impressions for group i
- R_{ij} is # views in group i per dollar spent on ads in channel j
- s_j is amount of advertising purchased in channel j
- $(Rs)_i$ is total number of views in group i
- least squares problem: minimize $\|Rs - v^{\text{des}}\|^2$ (ignoring $s \geq 0$ and budget)

Example: $m = 10$, $n = 3$, $v^{\text{des}} = 10^3 \times \mathbf{1}$





Solution

a solution must satisfy the *normal equations:*

$$A^T A x^\star = A^T b$$

if the columns of A are linearly independent, then the solution is unique:

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

- $A^\dagger = (A^T A)^{-1} A^T$ is the psuedo-inverse of A, which is also a left inverse
- solution is sometimes called the *least squares approximate solution* of $Ax = b$
  - $x^\star = A^\dagger b$ solves the linear equation $Ax = b$ if it has a solution ($b \in \mathrm{range}(A)$)
  - if $Ax = b$ does not have a solution, then $Ax^\star \neq b$
- any x satisfying the above is a global minimizer since $\nabla^2 f(x) = 2A^T A \geq 0$

MATLAB command

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

Example

we are given two different types of concrete:

first type	second type
30% cement	10% cement
40% gravel	20% gravel
30% sand	70% sand (all percentages of weight)

how many pounds of each type to mix so that you get a mixture close to

- 5 pounds of cement
- 3 pounds of gravel
- 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:

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

  where $x = (x_1, x_2)$

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

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

Optimality proof using algebra

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

using $A^TAx^\star = A^Tb$, the cross product term is zero; hence

$$\|Ax - b\|^2 = \|A(x - x^\star)\|^2 + \|Ax^\star - b\|^2$$

- since $\|A(x - x^\star)\|^2 \geq 0$, we have $\|Ax - b\|^2 \geq \|Ax^\star - b\|^2$
- if A has linearly independent columns, then

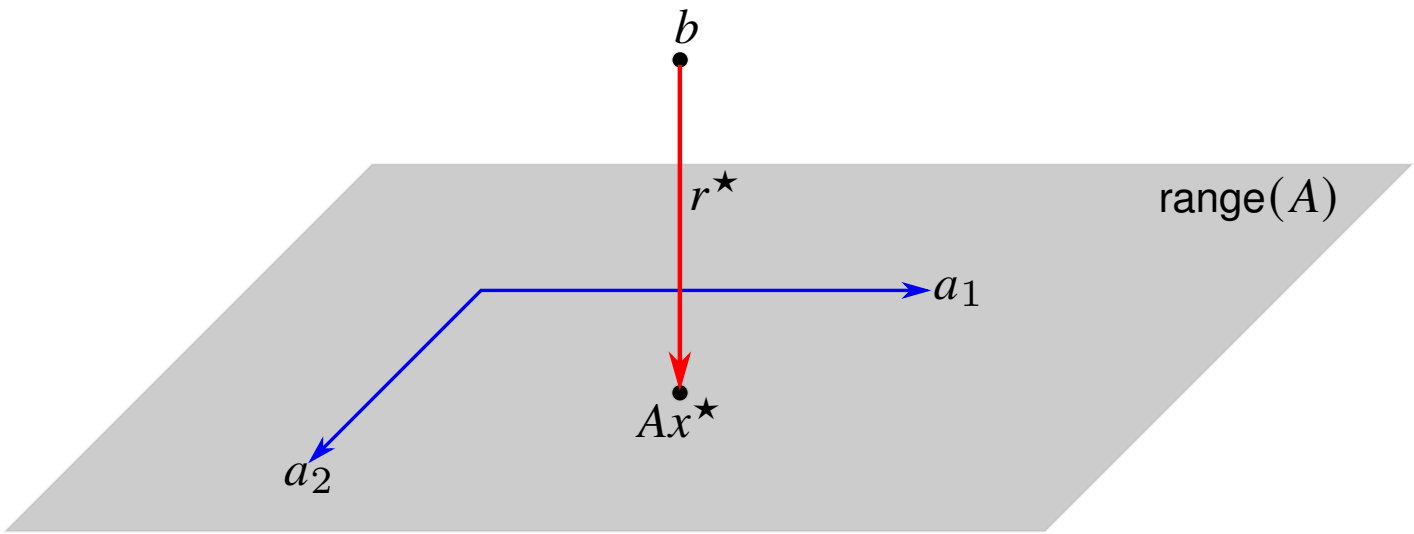
  $$\|Ax - b\|^2 > \|Ax^\star - b\|^2 \quad \text{(unique solution)}$$

  this is because $\|A(x - x^\star)\|^2 = 0 \Rightarrow A(x - x^\star) = 0 \Rightarrow x = x^\star$

Geometric interpretation

- $Ax^\star$ is the vector in range$(A) = \text{span}(a_1, \ldots, a_n)$ closest to b
- the optimal residual $r^\star = Ax^\star - b$ is orthogonal to range(A): $r^\star \perp Aw$ for any w

$$(Aw)^T r^\star = (Aw)^T(Ax^\star - b) = w^T A^T(Ax^\star - b) = w^T 0 = 0,$$



- $Ax^\star = AA^\dagger b$ is projection on range(A)

Data fitting

Setup: a scalar y and an p-vector z are related by model, $g : \mathbb{R}^p \to \mathbb{R}$,

$$y \approx g(z)$$

- z is the *independent variable* or *feature vector*
- y is the *outcome* or *response variable*
- we don't know g, which gives the 'true' relationship between z and y

Data: we are given some data (*observations*, *samples*, or *measurements*)

$$z^{(1)}, \ldots, z^{(m)}, \quad y^{(1)}, \ldots, y^{(m)}$$

- $(z^{(i)}, y^{(i)})$ is ith *data pair*
- $z_j^{(i)}$ is the jth component of ith data point $z^{(i)}$

Model fitting: find g or an approximation of it based on observations by minimizing

$$\frac{1}{m} \sum_{i=1}^{m} (g(z^{(i)}) - y^{(i)})^2 \tag{7.1}$$

Linear in parameters model fitting

Linear in parameters model

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

- $g_i(z)$ are *basis functions* or *feature mappings* that we choose
- x_i are *model parameters*
- we want to estimate x such that the cost (7.1) is minimized

Least squares formulation: minimize $\|Ax - b\|^2$ where

$$A = \begin{bmatrix} g_1(z^{(1)}) & g_2(z^{(1)}) & \cdots & g_n(z^{(1)}) \\ g_1(z^{(2)}) & g_2(z^{(2)}) & \cdots & g_n(z^{(2)}) \\ \vdots & \vdots & \cdots & \vdots \\ g_1(z^{(m)}) & g_2(z^{(m)}) & \cdots & g_n(z^{(m)}) \end{bmatrix}, \quad b = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

Polynomial fit

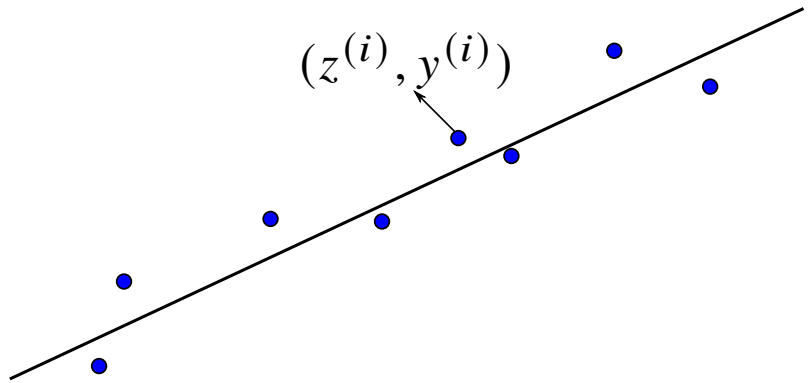
model is a polynomial of degree at most $n-1$:

$$g(z) = x_1 + x_2 z + \cdots + x_n z^{n-1}$$

- $g_i(x) = z^{i-1}$, $i = 1, \ldots, n$; here z^i means scalar z to ith power
- $z^{(i)}$ is ith data point
- A is Vandermonde matrix

$$A = \begin{bmatrix} 1 & z^{(1)} & \cdots & (z^{(1)})^{n-1} \\ 1 & z^{(2)} & \cdots & (z^{(2)})^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & z^{(m)} & \cdots & (z^{(m)})^{n-1} \end{bmatrix}$$

Line fitting



find a straight line that best fits the data $(z^{(i)}, y^{(i)})$:

$$x_1 + x_2 z \approx y$$

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

$$A = \begin{bmatrix} 1 & z^{(1)} \\ 1 & z^{(2)} \\ \vdots & \vdots \\ 1 & z^{(m)} \end{bmatrix}, \quad b = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

Example

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

$$\sum_{i=1}^{3} (x_1 + x_2 z^{(i)} - y^{(i)})^2 = (x_1 + 2x_2 - 3)^2 + (x_1 + 3x_2 - 4)^2 + (x_1 + 4x_2 - 15)^2$$

$$= \|Ax - b\|^2$$

where

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

- the solution is

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

the equation of the line is $g(z) = 6z - 32/3$

Regression

consider the *regression model*:

$$y \approx g(z) = z^T w + v$$

- least squares regression: choose the model parameters v, β that minimize

$$\frac{1}{m} \sum_{i=1}^{m} \left(v + (z^{(i)})^T w - y^{(i)} \right)^2 = \|Ax - y\|^2$$

with

$$A = \begin{bmatrix} 1 & (z^{(1)})^T \\ \vdots & \vdots \\ 1 & (z^{(m)})^T \end{bmatrix}, \quad x = \begin{bmatrix} v \\ w \end{bmatrix}, \quad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

- same as data fitting with basis functions

$$g_1(z) = 1, \; g_i(z) = z_{i-1}, \quad i = 2, \ldots, p+1$$

Linear estimation

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

$$y_t = a_t^T x + v_t, \quad t = 1, \ldots, m$$

- a_t^T are known or measured linear system parameters
- v_t is an unknown small measurement noise
- the estimation problem is to find a good x such that $y_t - a_t^T x$ is minimized for all t
- we can formulate this as a least square problem with

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}, \quad b = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Example

- n measurements of voltage across resistor with 1-ampere current

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

- we wish to find R that best fits our measurements

this problem can be formulated as

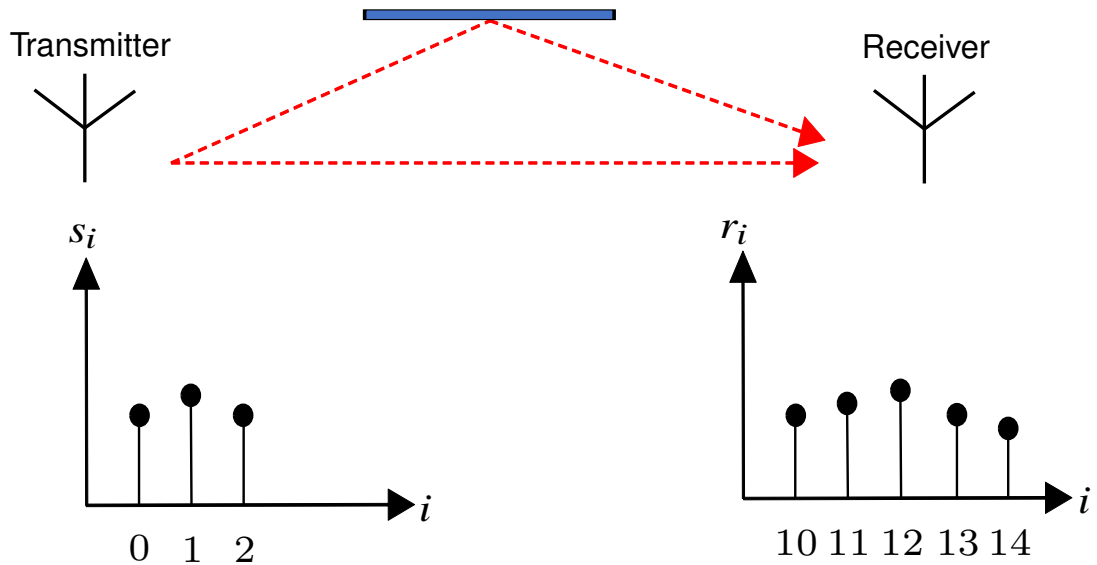
$$\text{minimize} \quad \left\| \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} R - \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \right\|^2$$

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

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

Example

- 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 $t = 10$ to $t = 14$, which is the sum of the signals from these two paths plus some unknown noise



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

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

we can formulate this as a least squares problem with

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

the least squares solution is

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

Outline

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

Regularized least squares

$$\text{minimize} \quad \|Ax - b\|^2 + \delta\|Rx\|^2$$

- $R \in \mathbb{R}^{p\times n}$ is the *regularization matrix*; δ is the *regularization parameter*
- large δ gives more emphasis on making the term $\delta\|Rx\|^2$ small

Why regularization?

- useful when A has linearly dependent columns
- utilize some prior information about x

Solution

$$(A^TA + \delta R^TR)x = A^Tb$$

if $A^TA + \delta R^TR$ is invertible, then

$$x^\star = (A^TA + \delta R^TR)^{-1}A^Tb$$

Example: signal de-noising

- $x = (x_1, x_2, \ldots, x_n)$ 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

$$y = x + v$$

  where v is some noise

- the signal does not vary too much $|x_{i+1} - x_i| << 1$
- given y, we want to find a "good" estimate of x

Naive solution

- directly set $x = y$
- this can result in a bad estimate if some noise components v_i are large

Least squares formulation

$$\text{minimize} \quad \|x - y\|^2 + \delta \|Rx\|^2$$

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

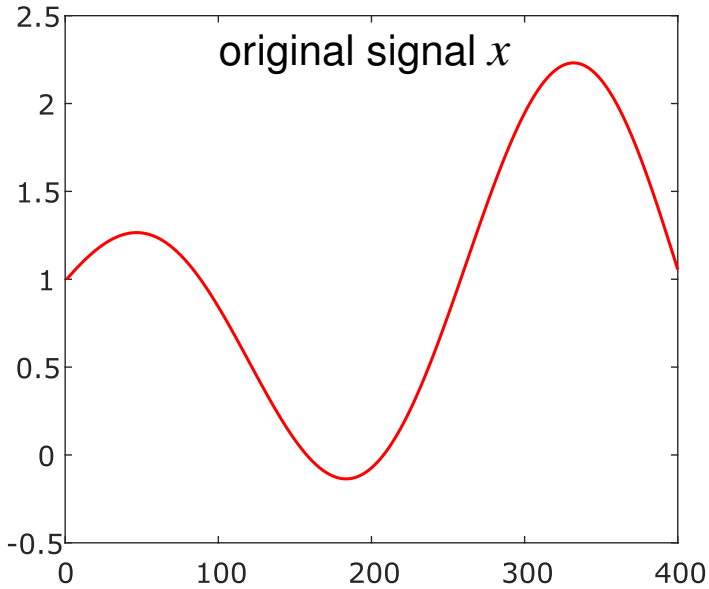
$$\|Rx\|^2 = \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2$$

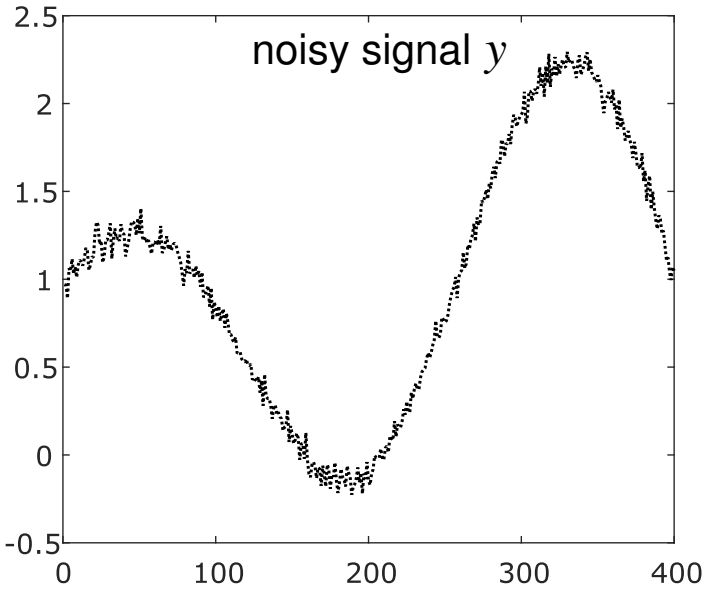
the matrix R is the difference matrix

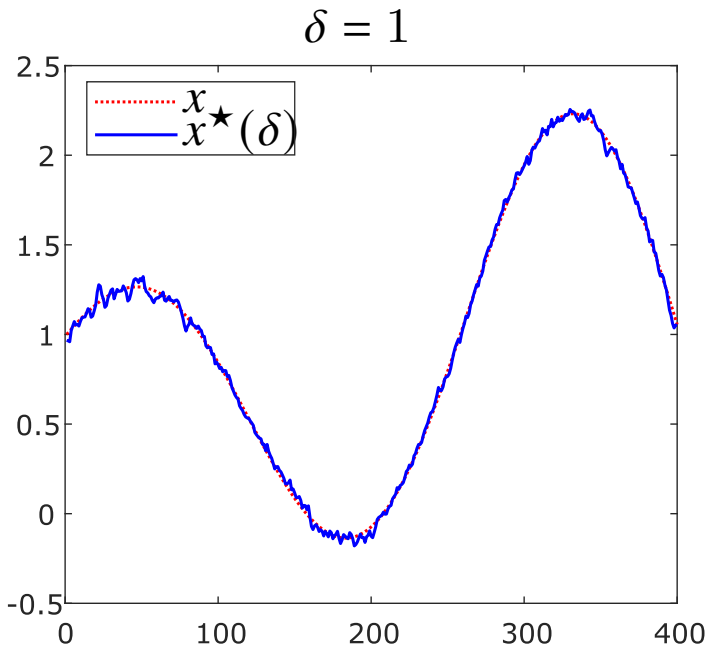
$$R = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

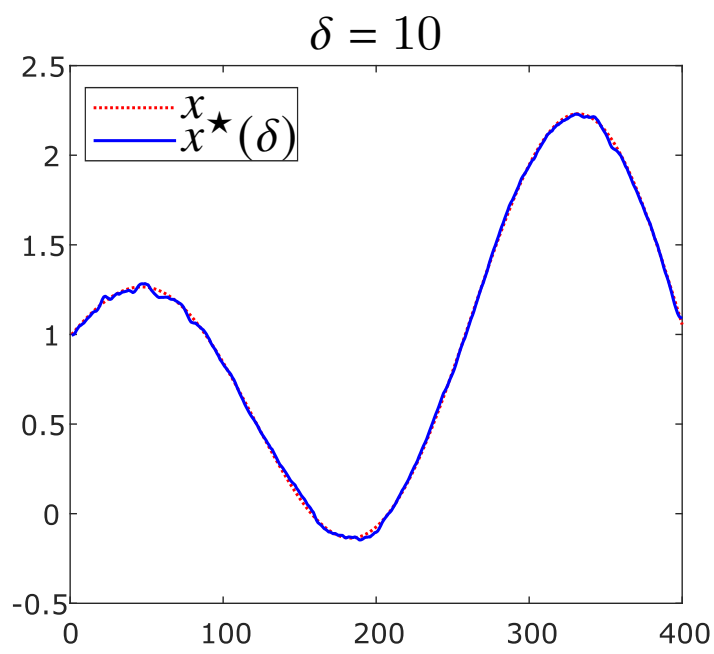
- the optimal solution is given by

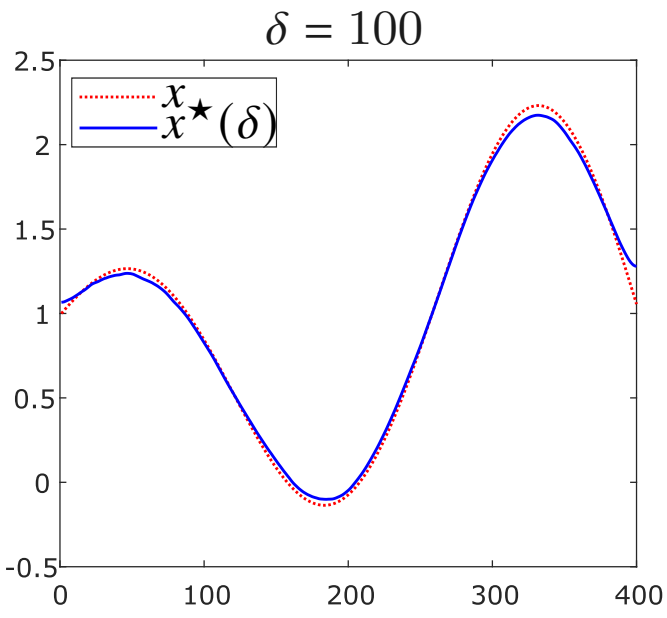
$$x^\star(\delta) = (I + \delta R^T R)^{-1} y$$

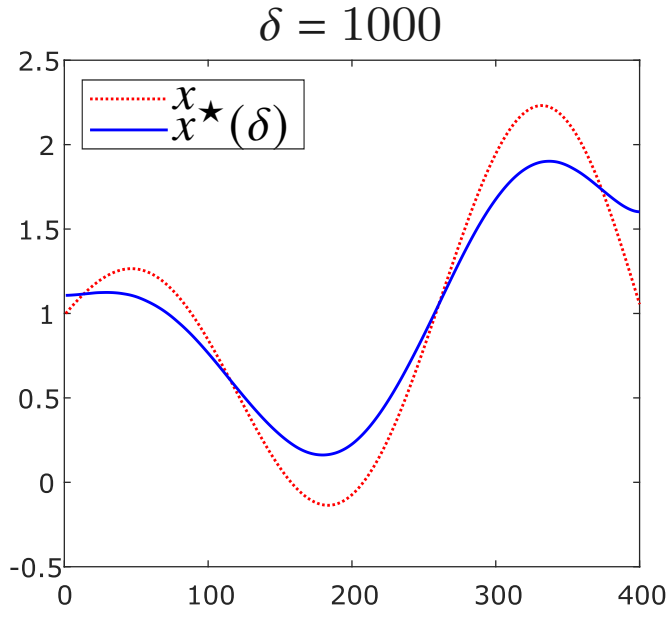

original signal x
2.5
2
1.5
1
0.5
0
-0.5
0
100
200
300
400


noisy signal y
2.5
2
1.5
1
0.5
0
-0.5
0
100
200
300
400


$\delta = 1$
x
$x^\star(\delta)$
2.5
2
1.5
1
0.5
0
-0.5
0
100
200
300
400


$\delta = 10$
x
$x^\star(\delta)$
2.5
2
1.5
1
0.5
0
-0.5
0
100
200
300
400


$\delta = 100$
x
$x^\star(\delta)$
2.5
2
1.5
1
0.5
0
-0.5
0
100
200
300
400


$\delta = 1000$
x
$x^\star(\delta)$
2.5
2
1.5
1
0.5
0
-0.5
0
100
200
300
400

Outline

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

Nonlinear least squares

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

- $r : \mathbb{R}^n \to \mathbb{R}^m$ is nonlinear function with components $r_i : \mathbb{R}^n \to \mathbb{R}$
- when $r(x) = Ax - b$, 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:

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

Location from distance of measurements

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

$$\gamma_i = \|x - y_i\| + v_i, \quad i = 1, \ldots, m$$

  where v_i is some small measurement noise

- we can estimate the position of x by solving

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

  this is a nonlinear least squares problem with $r_i(x) = \|x - y_i\| - \gamma_i$

Nonlinear data-fitting

Model fitting problem

- given m *data points/measurements* $(z^{(i)}, y^{(i)})$, $i = 1, \ldots, m$, $z_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$
- these points are approximately related by the equation

$$g(z; x) \approx y$$

  where $g : \mathbb{R}^p \to \mathbb{R}$ and $x \in \mathbb{R}^n$ are unknown parameters

Nonlinear least squares formulation

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

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

Example

- given m measurements, $y^{(1)}, \ldots, y^{(m)}$, at m times, $t^{(1)}, \ldots, t^{(m)}$ of:

$$y^{(i)} = \beta \sin(\omega t^{(i)} + \phi) + n(t^{(i)})$$

  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

$$\text{minimize} \quad \sum_{i=1}^{m} r_i(x)^2 = \sum_{i=1}^{m} \left(y^{(i)} - \beta \sin(\omega t^{(i)} + \phi)\right)^2$$

with variable $x = (\beta, \omega, \phi)$ and $r_i(x) = y^{(i)} - \beta \sin(\omega t^{(i)} + \phi)$

Classification

Classification

- m training data points $(z^{(1)}, y^{(1)}), \ldots, (z^{(m)}, y^{(m)})$
- outcome y_i takes on finite number of values called *labels* or *categories*
  - TRUE or FALSE
  - SPAM or NOT SPAM
  - DOG, HORSE, or MOUSE
- data fitting $g(z^{(i)}) \approx y^{(i)}$ is called *classification*
- model used to determine which class the a new data point z belongs to

Boolean (2-way) classification

- two possible outcomes only encoded as $y \in \{+1, -1\}$
- Boolean classifier has form $y = g(z)$, $g : \mathbb{R}^p \rightarrow \{-1, +1\}$

Least squares Boolean classifier

- we are given the data points $(z^{(i)}, y^{(i)})$, $i = 1, \ldots, m$
- determine basis functions $g_1, \ldots, g_n$ for linear-in-parameter model

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

- use least squares data-fitting to find parameters $x_1, \ldots, x_n$
- take the sign of $\tilde{g}(z)$ to get the *Boolean classifier:*

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

- often used with regression model $\tilde{g}(z) = z^T\beta + v$

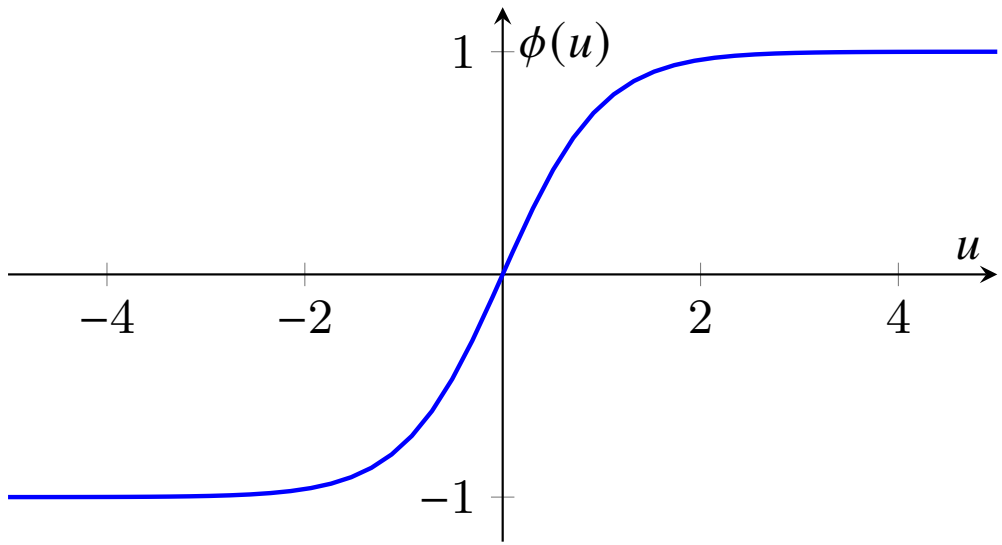
Nonlinear least squares classification

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

where $\phi : \mathbb{R} \longrightarrow \mathbb{R}$ is the sigmoidal function (hyperbolic tangent):

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

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



Outline

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

Linear least square approximation at each iteration

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

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

$$\hat{r}(x; x^{(k)}) = r(x^{(k)}) + Dr(x^{(k)})(x - x^{(k)})$$

  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

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

- the above problem is a linear least squares problem with

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

Gauss-Newton method

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

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

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

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

hence $x^{(k)}$ satisfies the optimality condition since $\nabla f(x) = 2Dr(x)^T r(x)$

Gauss-Newton algorithm

---

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

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

1. evaluate $Dr(x^{(k)}) = (\nabla r_1(x^{(k)})^T, \ldots, \nabla r_m(x^{(k)})^T)$
2. **if** $\|2Dr(x^{(k)})^T r(x^{(k)})\|^2 \leq \epsilon$ or $\|r(x^{(k)})\| < \epsilon$, stop and output $x^{(k)}$
3. set $x^{(k+1)}$ to be the minimizer of $\|Dr(x^{(k)})x - (Dr(x^{(k)})x^{(k)} - r(x^{(k)}))\|^2$:

$$x^{(k+1)} = x^{(k)} - (Dr(x^{(k)})^T Dr(x^{(k)}))^{-1} Dr(x^{(k)})^T r(x^{(k)})$$

- 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

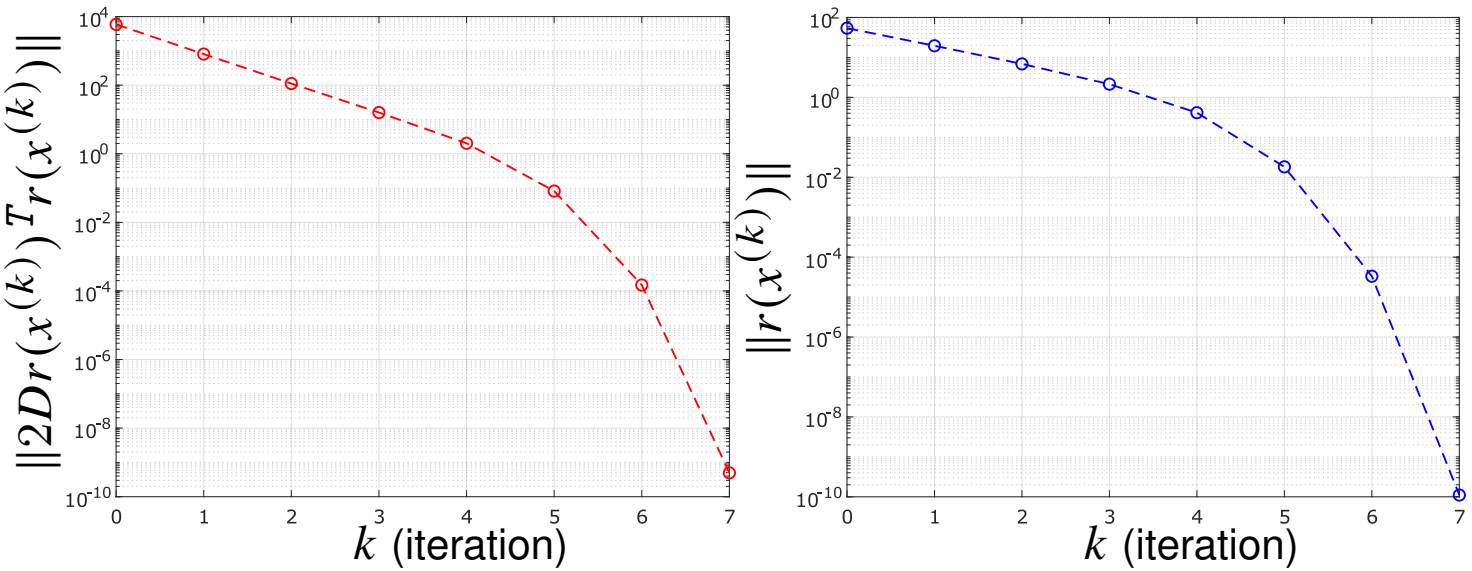
- approximation $\|r(x)\|^2 \approx \|\hat{r}(x; x^{(k)})\|^2$ holds when x near $x^{(k)}$
- when $x^{(k+1)}$ is not near $x^{(k)}$, the affine approximation will not be accurate
- so the algorithm may fail or diverge ($\|r(x^{(k+1)})\| > \|r(x^{(k)})\|$)
- a second major issue is that columns of the matrix $Dr(x^{(k)})$ may not always be linearly independent; in this case, the next iterate is not defined

Example

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

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

Example

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

where p_i, q_i, γ_i are given

the gradient of r_i is

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

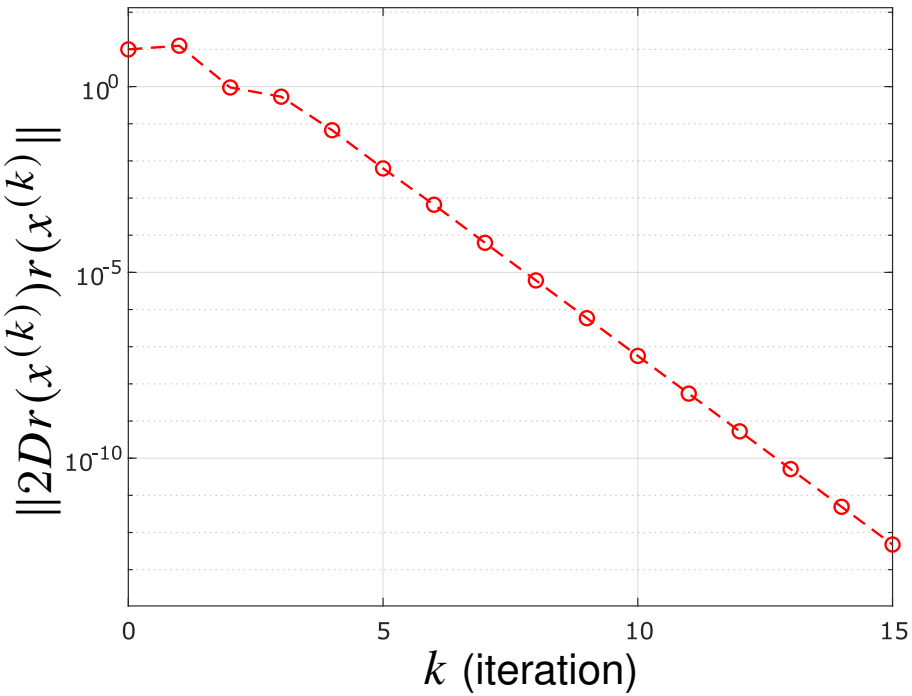
thus, the Jacobian of r is

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

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

results with data

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



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

Relation to Newton method for nonlinear equations

- Gauss-Newton update

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

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

$$x^{(k+1)} = x^{(k)} - Dr(x^{(k)})^{-1} r(x^{(k)})$$

Relation to Newton method for unconstrained minimization

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

- gradient:

$$\nabla f(x) = 2\sum_{i=1}^{m} r_i(x)\nabla r_i(x) = 2Dr(x)^T r(x)$$

- second derivatives:

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

- Hessian

$$\nabla^2 f(x) = 2Dr(x)^T Dr(x) + 2\sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)$$

Newton and Gauss-Newton steps

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

$$\begin{aligned} v_{\mathrm{nt}} &= -\nabla^2 f(x)^{-1} \nabla f(x) \\ &= -\Big(Dr(x)^T Dr(x) + \sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x)\Big)^{-1} Dr(x)^T r(x) \end{aligned}$$

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

$$v_{\mathrm{gn}} = -\Big(Dr(x)^T Dr(x)\Big)^{-1} Dr(x)^T r(x)$$

  can be written as $v_{\mathrm{gn}} = -H_{\mathrm{gn}}^{-1} \nabla f(x)$ where $H_{\mathrm{gn}} = Dr(x)^T Dr(x)$

- H_{gn} is the Hessian without the term $\sum_i r_i(x) \nabla^2 r_i(x)$

Comparison

Newton step

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

Gauss-Newton step

- Gauss-Newton iteration is cheaper (does not require second derivatives)
- a descent direction (if columns of $Dr(x)$ are linearly independent):

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

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

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

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

Outline

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

Regularized approximate problem

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

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

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

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

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

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

Levenberg-Marquardt update

the solution is

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

Updating δ

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

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

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

Levenberg-Marquardt algorithm

---

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

repeat for $k \geq 0$

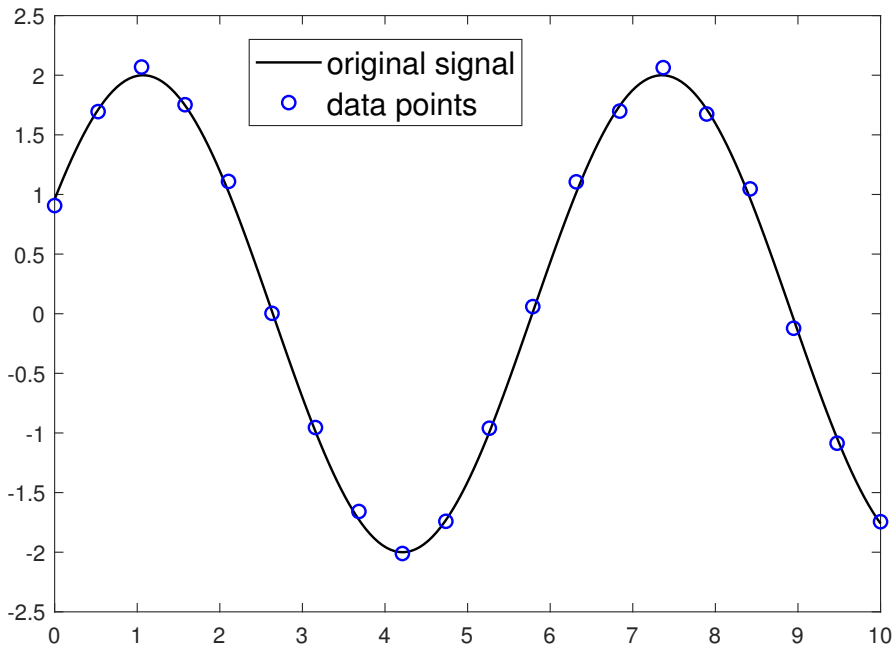
1. evaluate $Dr(x^{(k)}) = (\nabla r_1(x^{(k)})^T, \ldots, \nabla r_m(x^{(k)})^T)$.
2. **if** $\|2Dr(x^{(k)})^T r(x^{(k)})\|^2 \leq \epsilon$ or $\|r(x^{(k)})\| < \epsilon$, stop and output $x^{(k)}$
3. set $x^{(k+1)}$ to be the minimizer of

$$\|r(x^{(k)}) + Dr(x^{(k)})(x - x^{(k)})\|^2 + \delta_k \|x - x^{(k)}\|^2$$

4. if $\|r(x^{(k+1)})\|^2 < \|r(x^{(k)})\|^2$, then decrease δ_{k+1} (*e.g.*, $\delta_{k+1} = 0.8\delta_k$); otherwise, increase δ_{k+1} (*e.g.*, $\delta_{k+1} = 2\delta_k$) and set $x^{(k+1)} = x^{(k)}$

Example

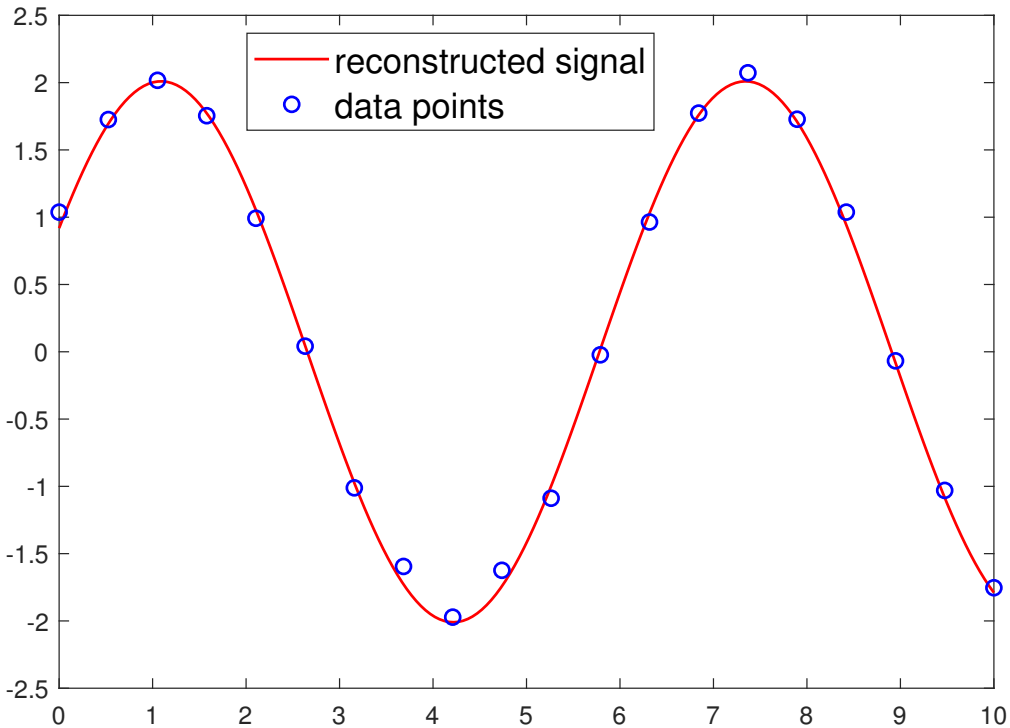
- data-fitting problem with $r_i(\beta, \omega, \phi) = y^{(i)} - \beta \sin(\omega t^{(i)} + \phi)$
- find (β, ω, ϕ) given $m = 20$ data points



- 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 results in



References and further readings

- S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares.* Cambridge University Press, 2018. (chapters 12, 13, 14, 15, 18)
- L. Vandenberghe, *EE133A Lecture Notes*, UCLA.
- E. K.P. Chong, Wu-S. Lu, and S. H. Zak. *An Introduction to Optimization: With Applications to Machine Learning*. John Wiley & Sons, 2023. (chapter 12)
- A. Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with Python and MATLAB.* SIAM, 2023. (chapter 3)