Introduction to Overdetermined Linear Systems

Contents

Opening Example: Polynomial Approximation

Introduction

Problem: Fitting Functions to Data

Given data points $\{(x_i, y_i) \mid i \in \mathbb{N}[1, m]\}$, pick a form for the "fitting" function $f(x)$ and minimize its total error in representing the data.

- With real-world data, interpolation is often not the best method.
- Instead of finding functions lying exactly on given data points, we look for ones which are "close" to them.
- In the most general terms, the fitting function takes the form

$$f(x) = c_1 f_1(x) + \cdots + c_n f_n(x),$$

where $f_1, \ldots, f_n$ are known functions while $c_1, \ldots, c_n$ are to be determined.

Linear Least Squares Approximation

In this discussion:

- use a polynomial fitting function $p(x) = c_1 + c_2 x + \cdots + c_n x^{n-1}$ with $n < m$;
- minimize the 2-norm of the error $r_i = y_i - p(x_i)$:

$$\|\mathbf{r}\|_2 = \sqrt{\sum_{i=1}^{m} r_i^2} = \sqrt{\sum_{i=1}^{m} \big(y_i - p(x_i)\big)^2}.$$

Since the fitting function is linear in unknown coefficients and the 2-norm is minimized, this method of approximation is called the **linear least squares (LLS) approximation**.

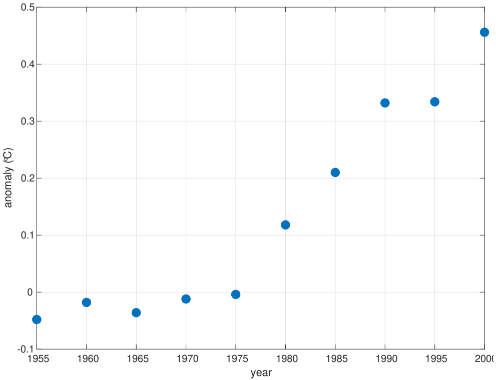
Example: Temperature Anomaly

Below are 5-year averages of the worldwide temperature anomaly as compared to the 1951-1980 average (source: NASA).

Year	Anomaly ($^\circ C$)
1955	-0.0480
1960	-0.0180
1965	-0.0360
1970	-0.0120
1975	-0.0040
1980	0.1180
1985	0.2100
1990	0.3320
1995	0.3340
2000	0.4560

Import and Plot Data

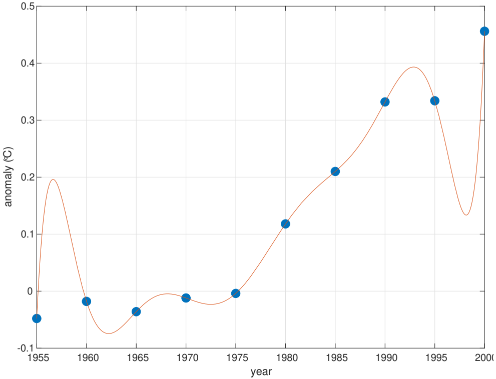
```
t = (1955:5:2000)';
y = [-0.0480; -0.0180;
     -0.0360; -0.0120;
     -0.0040;  0.1180;
      0.2100;  0.3320;
      0.3340;  0.4560];
plot(t, y, '.')
```



Example: Temperature Anomaly (cont')

Interpolation

```
t = (t-1950)/10;
n = length(t);
V = t.^(0:n-1);
c = V\y;
p = @(x) polyval(flip(c),
      (x-1950)/10);
hold on
fplot(p, [1955 2000])
```



Fitting by a Straight Line

Suppose that we are fitting data to a linear polynomial: $p(x) = c_1 + c_2 x$.

- If it were to pass through all data points:

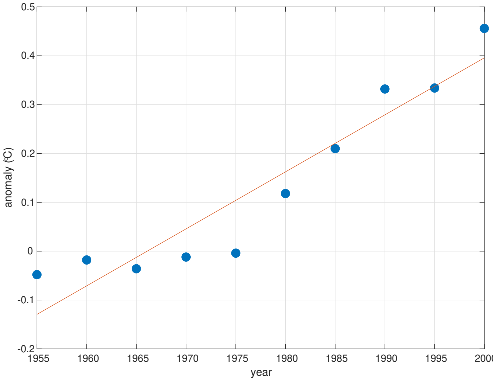
$$\begin{cases} y_1 = p(x_1) = c_1 + c_2 x_1 \\ y_2 = p(x_2) = c_1 + c_2 x_2 \\ \quad \vdots \qquad\quad \vdots \qquad \vdots \\ y_m = p(x_m) = c_1 + c_2 x_m \end{cases} \xrightarrow{\text{matrix equation}} \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}}_{\mathbf{c}}$$

- The above is unsolvable; instead, find $\mathbf{c}$ which makes the *residual* $\mathbf{r} = \mathbf{y} - V\mathbf{c}$ "as small as possible" in the sense of vector 2-norm.
- **Notation:** $\mathbf{y}$ "=" $V\mathbf{c}$

Fitting by a Striaght Line: MATLAB Implementation

Revisiting the temperature anomaly example again:

```
year = (1955:5:2000)';
t = year - 1955;
V = t.^(0:1);
c = V\y;
p = @(x) polyval(flip(c),
      x-1955);
plot(year, y, '.')
hold on
fplot(p, [1955, 2000])
```



Fitting by a General Polynomial

In general, when fitting data to a polynomial

$$p(x) = c_1 + c_2 x + c_3 x^2 + \cdots + c_n x^{n-1},$$

we need to solve

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{\mathbf{y}} \text{“="} \underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{n-1} \end{bmatrix}}_{V} \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}}_{\mathbf{c}} .$$

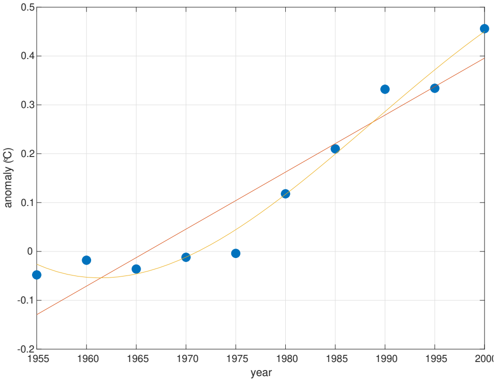
- The solution $\mathbf{c}$ of $\mathbf{y}$ “=" $V\mathbf{c}$ turns out to be the solution of the *normal equation*

$$V^{\mathrm{T}} V \mathbf{c} = V^{\mathrm{T}} \mathbf{y} .$$

Fitting by a General Polynomial: MATLAB Implementation

Revisiting the temperature anomaly example again:

```
V = t.^(0:3);
c = V\y;
q = @(x) polyval(flip(c),
     x-1955);
hold on
fplot(q, [1955 2000])
```



Backslash Again

The Versatile Backslash

In MATLAB, the generic linear equation $A\mathbf{x} = \mathbf{b}$ is solved by `x = A\b`.

- When A is a square matrix, Gaussian elimination is used.
- When A is NOT a square matrix, the normal equation $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$ is solved instead.

- As long as $A \in \mathbb{R}^{m\times n}$ where $m \geqslant n$ has rank n, the square matrix $A^{\mathrm{T}}A$ is nonsingular. (unique solution)
- Though $A^{\mathrm{T}}A$ is a square matrix, MATLAB does not use Gaussian elimination to solve the normal equation.
- Rather, a faster and more accurate algorithm is used.

The Normal Equations

LLS and Normal Equation

Big Question: How is the least square solution $\mathbf{x}$ to $A\mathbf{x}$ "=" $\mathbf{b}$ equivalent to the solution of the normal equation $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$?

Theorem (Normal Equation)

Let $A \in \mathbb{R}^{m \times n}$ with $m \geqslant n$. If $\mathbf{x} \in \mathbb{R}^n$ satisfies $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$, then $\mathbf{x}$ solves the LLS problems, *i.e.*, $\mathbf{x}$ minimizes $\|\mathbf{b} - A\mathbf{x}\|_2$.

- **Idea of Proof**. Enough show to that $\|\mathbf{b} - A(\mathbf{x} + \mathbf{y})\|_2 \geqslant \|\mathbf{b} - A\mathbf{x}\|_2$ for any $\mathbf{y} \in \mathbb{R}^n$.
- **Useful identity**.

$$\|\mathbf{u} \pm \mathbf{v}\|_2^2 = \|\mathbf{u}\|_2^2 + \|\mathbf{v}\|_2^2 \pm 2\mathbf{u}^{\mathrm{T}}\mathbf{v}, \qquad (\star)$$

Proof of the Theorem

Proof. Let $\mathbf{y} \in \mathbb{R}^m$ be arbitrary. Using the identity $(\star)$, we can write

$$\left\|\mathbf{b} - A(\mathbf{x} + \mathbf{y})\right\|_2^2 = \|\mathbf{b} - A\mathbf{x}\|_2^2 + \|A\mathbf{y}\|_2^2 - 2\mathbf{y}^{\mathrm{T}} A^{\mathrm{T}} (\mathbf{b} - A\mathbf{x}).$$

Since $\mathbf{x}$ solves the normal equation $A^{\mathrm{T}} A\mathbf{x} = \mathbf{b}$, the last term vanishes; since $\|A\mathbf{y}\|_2 \geqslant 0$, it follows that

$$\left\|\mathbf{b} - A(\mathbf{x} + \mathbf{y})\right\|_2^2 \geqslant \|\mathbf{b} - A\mathbf{x}\|_2^2 .$$

Since $\mathbf{y}$ was chosen arbitrarily, this shows that $\mathbf{x}$ minimizes $\|\mathbf{b} - A\mathbf{x}\|$. $\square$

Appendix: Derivation of Normal Equation

Derivation of Normal Equation

Consider $A\mathbf{x}$ "=" $\mathbf{b}$ where $A \in \mathbb{R}^{m\times n}$ where $m \geqslant n$.

- **Requirement:** minimize the 2-norm of the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$:

$$g(x_1, x_2, \ldots, x_n) := \|\mathbf{r}\|_2^2 = \sum_{i=1}^{m} \left(b_i - \sum_{j=1}^{n} a_{ij} x_j\right)^2 .$$

- **Strategy:** using calculus, find the minimum by setting

$$\mathbf{0} = \nabla g(x_1, x_2, \ldots, x_n)$$

  which yields n equations in n unknowns $x_1, x_2, \ldots, x_n$.

Derivation of Normal Equation (cont')

Noting that $\partial x_j / \partial x_k = \delta_{j,k}$, the n equations $\partial g / \partial x_k = 0$ are written out as

$$0 = \sum_{i=1}^{m} 2\Big(b_i - \sum_{j=1}^{n} a_{ij} x_j\Big)(-a_{ik}), \qquad \text{for } k \in \mathbb{N}[1, n]\,,$$

which can be rearranged into

$$\sum_{i=1}^{m} a_{ik} b_i = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} a_{ik} x_j, \qquad \text{for } k \in \mathbb{N}[1, n]\,.$$

One can see that the two sides correspond to the k^{th} elements of $A^{\mathrm{T}}\mathbf{b}$ and $A^{\mathrm{T}}A\mathbf{x}$ respectively:

$$A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}\,,$$

showing the desired equivalence.