Lineare Ausgleichsrechnung

1. Ausgleichsgeraden

(1.1) Problemstellung

Finde zu gegebenen Daten $(x_j, y_j) \in \mathbb{R} \times \mathbb{R}$ $j = 1..N$

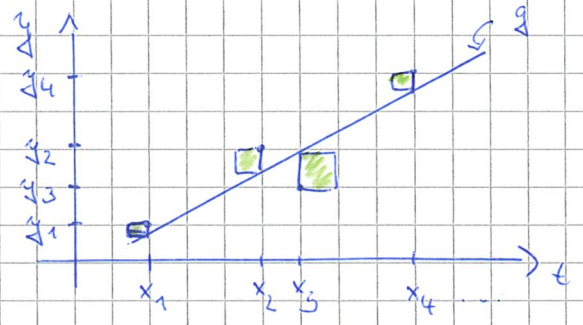
eine Gerade $x \mapsto g(x) = mx + c$ so, dass

$$g(x_j) \approx y_j \quad \text{für } j = 1..N$$

Genauer: Finde m und c so, dass

$$\sum_{j=1}^{N} |y_j - g(x_j)|^2 \quad \text{minimal wird}$$

(1.2) Geometrische Interpretation



Minimiere die Fläche

"Methode der kleinsten Fehlerquadrate" (Gauß 1803)

(1.3) Matrix - Vektor - Formulierung

Finde $\begin{pmatrix} m \\ c \end{pmatrix} \in \mathbb{R}^2$ so, dass

$$\| A \begin{pmatrix} m \\ c \end{pmatrix} - Y \|_2^2 \quad \text{minimal wird}$$

mit

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_N & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

$$\| Y \|_2^2 = \sum_{j=1}^{N} |y_j|^2 \qquad \| \|_2 \text{ euklidische Norm}$$

2 Multiple lineare Regression

(2.1) Problemstellung

Finde zu Daten $(X_{j1}, \dots X_{jn}, Y_j)$ $j = 1 \dots m$

($m \gg n$) Parameter $\beta_1 \dots \beta_n$ so, dass

$$Y_j \approx \sum_{i=1}^{n} \beta_i X_{ji}$$

genauer $\sum_{j=1}^{m} \left(Y_j - \sum_{i=1}^{n} \beta_i X_{ji} \right)^2$ so minimal werden (Warum / Wann das sinnvoll ist $\rightsquigarrow$ Ang St / Stochastik)

(2.2) Matrix - Vektor - Formulierung

Finde $\beta \in \mathbb{R}^n$ so, dass $\| X\beta - Y \|_2^2 = \min!$

für $X = \begin{bmatrix} X_{11} & & X_{1n} \\ & & \\ X_{m1} & & X_{mn} \end{bmatrix}$ $Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_m \end{bmatrix}$

(2.3) Verallgemeinerung

Finde zu Daten $(X_{j1}, \dots, X_{jn}, Y_j)$ $j = 1 \dots m$

und Funktionen φ_i $i = 1 \dots n$ Parameter $\beta_1 \dots \beta_n$

so, dass

$$Y_j \approx \sum_{i=1}^{n} \beta_i \varphi_i (X_{ji})$$

Dahinter steckt die Annahme dass die gemessenen Daten Y_j sich durch folgenden Zusammenhang aus den X_{ji} erklären lassen

$$y(x_1 \dots x_n) = \sum_{i=1}^{n} \beta_i \varphi_i (x_i)$$

Matrix - Vektor - Formulierung

$$\min_{\beta \in \mathbb{R}^n} \| A\beta - Y \|_2^2$$

$$A = (a_{ji})_{j,i=1}^{m\ n} \qquad a_{ji} = \varphi_i (X_{ji})$$

(2.4) Beispiel

Finde Polynom vom Grad $\leq n$ p, d.h. die Koeffizienten, so dass

für vorgegebene x_j, y_j $j = 1 .. m$ gilt

$$p(x_j) \approx y_j$$

$$\| \\ \sum_{i=0}^{n} \beta_i x_j^i$$

genauer: $\sum_{j=1}^{m} \left(\left(\sum_{i=0}^{n} \beta_i x_j^i \right) - y_j \right)^2 = \min!$

Matrix - Vektorformulierung

$$\min_{\beta \in \mathbb{R}^{n+1}} \| A\beta - y \|_2^2$$

$$A = \begin{bmatrix} 1 & x_1 & \cdots & x_1^n \\ | & & & \\ 1 & x_m & \cdots & x_m^n \end{bmatrix} = (a_{ij})_{\substack{j=1..m \\ i=0..n}}$$

$$a_{ij} = x_i^j$$

(2.5) Satz

Sei $A \in \mathbb{R}^{m \times n}$ $m \gg n$ mit $\operatorname{rang}(A) = n$

$b \in \mathbb{R}^n$. Das lineare Ausgleichsproblem

$\min_{x \in \mathbb{R}^n} \| Ax - b \|_2^2$ hat immer eine Lösung

Diese Lösung ist die Lösung der (linearen) "Normalengleichung"

$$A^T A x = A^T b$$

Bemerkung Die Normalengleichung ist i.a. schlecht konditioniert

Beweis $\rightsquigarrow$ Numerik 1

3) Lösung des linearen Ausgleichsproblems mit QR-Zerlegung

$A \in \mathbb{R}^{m \times n}$ $\quad m \geq n$ $\quad \operatorname{rang}(A) = n$

$b \in \mathbb{R}^m$

$$\min_{x \in \mathbb{R}^n} \| Ax - b \|_2^2$$

Sei $A = QR$ QR-Zerlegung von A dann gilt

$$\| Ax - b \|_2^2 = \| QRx - b \|_2^2 = \| Q (Rx - Q^T b) \|_2^2$$

$$= \| Rx - Q^T b \|_2^2 = \left\| \begin{bmatrix} \tilde{R} \\ 0 \end{bmatrix} x - \begin{bmatrix} \tilde{c} \\ \tilde{d} \end{bmatrix} \right\|_2^2$$

$$= \| \tilde{R} x - \tilde{c} \|_2^2 + \| \tilde{d} \|_2^2$$

für $\tilde{R} = (r_{ij})_{i,j=1}^{n}$ rechte obere Dreiecksmatrix

$\tilde{c}$: ersten n Einträge von $Q^T b$

$\tilde{d}$: Rest von $Q^T b$

$\Rightarrow$ $\| Ax - b \|_2^2$ wird minimal falls $x = \tilde{R}^{-1} \tilde{c}$

Algorithmus

Gegeben A, b

Rückgabe x^* Lösung von $\| Ax^* - b \|_2 = \min_{x \in \mathbb{R}} \| Ax - b \|_2$

Berechne QR-Zerlegung von A

Löse $\tilde{R} x = \tilde{c}$ $\quad \tilde{c} = (Q^T b)[0:n]$

$\uparrow$ gesuchtes x^*

Andere Lösungsverfahren für lineare Ausgleichsprobleme:

scipy.linalg.lstsq (engl. linear least squares)

numpy.linalg.lstsq