

Lineares Ausgleichsproblem:

Daten: $\begin{array}{c|c|c|c|c} t_i & 0 & 1 & 2 & 3 \\ \hline y_i & 3 & 2,14 & 1,8 & 1,72 \end{array}$ Modellfunktion:
 $y(t) = \alpha^{1/(1+t)} + \beta$

$$A = \begin{pmatrix} y(t_1=0) \\ y(1) \\ y(2) \\ y(t_n=3) \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix}; b = \begin{pmatrix} y_1=3 \\ 2,14 \\ 1,86 \\ y_n=1,72 \end{pmatrix}$$

$$\rightsquigarrow \begin{pmatrix} 1 & 1 \\ 1/2 & 1 \\ 1/3 & 1 \\ 1/4 & 1 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} y_1=3 \\ 2,14 \\ 1,86 \\ y_n=1,72 \end{pmatrix}$$

Satz: 1.1:

$x^* \in \mathbb{R}$ ist genau dann eine Lösung des linearen Ausgleichsproblems, wenn x^* Lösung der Normalgleichung $A^T A x = A^T b$ ist. Es gibt mindestens eine Lösung x^* . Sie ist eindeutig, gdw. $\text{Rang}(A) = n$.

Satz: 1.2:

Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ mit QR-Zerlegung von A , $\text{Rang}(A) \equiv n$, $A = QR$, $R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$, $R_1 \in \mathbb{R}^{n \times n}$

und $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} := Q^T b$ mit $c_1 \in \mathbb{R}^n$, $c_2 \in \mathbb{R}^{m-n}$.

Dann gilt:

R_1 ist regulär und

Außerdem gilt: $\|b - Ax\|_2 = \|c_2\|$.

Hessenbergmatrix durch**Householder-Reflexion:**

Mithilfe einer Householder-Reflexion, dargestellt durch Matrixmultiplikation $Q_u \cdot A$, kann ein Teil der Matrix zu null transformiert werden.

Mit v als Spaltenvektor von A , welcher die erste Spalte enthält, wird

$u := v + \text{sgn}(v_1) \cdot e_1 \cdot \|v\|$ gewählt ($\text{sgn}(\cdot)$ ist die Vorzeichenfunktion, jedoch muss bei 0 nicht 0 genommen werden!). Damit wird

$Q_u := \mathbb{1}_{m \times m} - 2 \cdot (u \cdot u^T) / (u^T \cdot u)$ definiert, welche A so verdreht, dass alle Elemente in der ersten Spalte unterhalb der Diagonalen verschwinden.

Nun kann man weiter vorgehen und die Teilmatrix von A hernehmen, welche die erste Zeile und Spalte gestrichen hat und darauf weiter agieren. Am ende hätte man mindestens eine obere rechte Dreiecksmatrix. Das Produkt aller verwendeten Q wäre dann eine orthogonale Matrix, womit $Q \cdot R = A$ als QR-Zerlegung entstanden ist.

Eigenschaften der Householder-Reflexion:

- (i) $Q_v \cdot v = -v$
- (ii) $Q_v \cdot u = u \Leftrightarrow v \perp u$
- (iii) $Q_v^T = Q_v^{-1} \Rightarrow Q_v$ ist Orthogonal

Eine **Givensrotation** von $A = \begin{pmatrix} a_1 & * \\ a_2 & * \end{pmatrix}$ kann mit

$$r = \sqrt{a_1^2 + a_2^2}, c = a_1/r, s = a_2/r \text{ und } G = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$

$$\text{erfolgen: } G \cdot A = \begin{pmatrix} r & * \\ 0 & * \end{pmatrix}$$

Nichtlineares Ausgleichsproblem:

Daten: $\begin{array}{c|c|c|c|c} t_i & 0 & 1 & 2 \\ \hline y_i & 2 & -3 & 4 \end{array}$ Modellfunktion:

$$y(t) = x_1 \sin(x_2 \cdot t)$$

$$\varphi(t, x) = y(t)$$

$$F_i = y_i - \varphi(t_i, x)$$

$$\mathbb{R}^n \ni F(x) = \begin{pmatrix} 2 - 0 \\ -3 - x_1 \sin(x_2) \\ 4 - x_1 \sin(2x_2) \end{pmatrix}$$

$$J_F(x) = \begin{pmatrix} 0 & 0 \\ -\sin(x_2) & -x_1 \cos(x_2) \\ -\sin(2x_2) & -2x_1 \cos(2x_2) \end{pmatrix}$$

Algorithmus: Gauß-Newton-Verfahren

- 1: Wähle Startvektor $x^{(0)} \in \mathbb{R}$
- 2: **for** $k = 0, 1, \dots$ **do**
 - ▷ Löse LGS nach $\Delta x^{(k)}$
- 3: $\|J_F(x^{(k)})\Delta x^{(k)} + F(x^{(k)})\|_2^2 \rightarrow \min$
- 4: setze $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$
- 5: **end for**

$A \in \mathbb{R}^{n \times n}$ s.p.d $\Rightarrow \exists V \in \mathbb{R}^{n \times n}$ orthogonal mit $V^T A V = D$; $d_{ii} = \lambda_i \geq 0 \Rightarrow A = V D V^T$ ($V V^T = \mathbb{1}$)

$$AV = VD$$

$A \in \mathbb{R}^{m \times n}$ $A^T A$ ist s.p.semi-d. $x^T A^T A x \geq 0$

Satz: Singulärwertzerlegung:

Sei $A \in \mathbb{R}^{m \times n}$ und $p = \min(m, n)$.

Dann existieren orthogonale Matrizen $U \in \mathbb{R}^{m \times m}$ und $V \in \mathbb{R}^{n \times n}$ mit

$$U^T A V = \Sigma = \text{diag}(\sigma_1, \dots, \sigma_p) \Rightarrow A = U \Sigma V^T$$

Anwendung der SVD

- (i) $A^{(s)} := \sum_{i=1}^s \sigma_i u_i v_i^T$, $s < r$ ist die beste (im Sinne der $\|\cdot\|_2$) Approximation von A mit dem Rang(s).
- (ii) **Pseudoinverse** A^+ , $A \in \mathbb{R}^{m \times n}$;
 $A = U \Sigma V^T$;
 $A^+ = V \Sigma^+ U^T$ mit
 $\Sigma^+ = \begin{pmatrix} 1/\sigma_1 & 0 & 0 \\ 0 & 1/\sigma_r & 0 \\ 0 & 0 & 0 \end{pmatrix}$;
 $x = A^+ b$ ist die Lösung des linearen Ausgleichsproblems $\|Ax - b\|_2 \rightarrow \min$
- (iii) Lösung des linearen Ausgleichsproblems für $\text{Rang}(A) < n$
- (iv) Regularisierung schlecht gestellter Probleme
 $\hat{x} = \sum_{i=1}^s 1/\sigma_i v_i \cdot (u_i^T \cdot b)$, $s < n$

Algorithmus: Vektoriteration

- 1: wähle $x^{(0)} \in \mathbb{R}$, setze $y^{(0)} = x^{(0)} / \|x^{(0)}\|_2$
- 2: **for** $k = 0, 1, \dots$ **do**
- 3: $x^{k+1} = A \cdot y^{(k)}$
- 4: $\lambda^{(k)} = y^{(k)T} \cdot x^{(k)}$
- 5: $y^{k+1} = x^{(k+1)} / \|x^{(k+1)}\|_2$
- 6: **end for**

Algorithmus: Inverse Vektoriteration mit Spektralverschiebung

- 1: wähle $x^{(0)} \in \mathbb{R}$, setze $y^{(0)} = x^{(0)} / \|x^{(0)}\|_2$
- 2: **for** $k = 0, 1, \dots$ **do**
- 3: Löse LGS $(A - \mu \mathbb{1}) x^{(k+1)} = y^{(k)}$
 $\triangleright \Leftrightarrow x^{(k+1)} = (A - \mu \mathbb{1})^{-1} \cdot y^{(k)}$
- 4: $\lambda^{(k+1)} = 1/y^{(k)T} \cdot x^{(k)} + \mu$
- 5: $y^{k+1} = x^{(k+1)} / \|x^{(k+1)}\|_2$
- 6: **end for**

Lemma zum QR-Verfahren

- (i) die Matrizen A_k sind **ähnlich** zu A .
 - (ii) A Symmetrisch $\Rightarrow A_k$ Symmetrisch.
 - (iii) A tridiagonal und Symmetrisch $\Rightarrow A_k$ auch.
- Zwei Matrizen A und B sind sich **ähnlich**, wenn es eine reguläre Matrix S gibt, so dass:
 $B = S^{-1} \cdot A \cdot S \Leftrightarrow SB = AS$ gilt.
- Ähnliche Matrizen haben dasselbe Spektrum
 $\sigma(A) = \sigma(B)$, gleiche Spur, gleichen Rang aber nicht notwendigerweise die gleichen EV.

Algorithmus: QR-Verfahren mit Spektralverschiebung

- 1: $A_0 = P^T \cdot A \cdot P$
 \triangleright Tridiagonaltransformation
- 2: **for** $k = 0, 1, \dots$ **do**
- 3: wähle $\mu_k \in \mathbb{R}$
- 4: $A_k - \mu_k \mathbb{1} = Q_k \cdot R_k$
 \triangleright QR-Zerlegung
- 5: $A_{k+1} = R_k \cdot Q_k + \mu_k \mathbb{1}$
 $\triangleright = Q_k^T A_k Q_k$
- 6: **end for**

Lagrange-Fundamental-Polynom

Seien Stützpunkte (x_i, f_i) $i = 0, \dots, n$ gegeben:

Suche Polynome $\ell_j \in \Pi_n$, $j = 0, \dots, n$ mit

$$\ell_j(x_k) = \begin{cases} 1 & : j = k \\ 0 & : j \neq k \end{cases}$$

$$P(f|x_0, \dots, x_n)(x) \stackrel{?!}{=} \sum_{j=0}^n f_j \cdot \ell_j(x);$$

$$\ell_j = \prod_{\substack{i=0 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i}$$

Newton-Basis:

$$P(f|x_0, \dots, x_n)(x) = \sum_{i=0}^n b_i \omega_i(x) = \sum_{i=0}^n b_i \prod_{p=0}^i (x - x_p)$$

$$b_k = f_{[x_0, \dots, x_k]}$$

$$f_{[x_r, \dots, x_s]} = (f_{[x_{r+1}, \dots, x_s]} - f_{[x_r, \dots, x_{s-1}]}) / (x_r - x_s)$$

$$\omega_j(x) = \prod_{i=0}^j (x - x_i), \omega_0(x) = 1$$

Algorithmus: Horner-Schema

- 1: $p = b_n$
- 2: **for** $k = n - 1, \dots, 1, 0$ **do**
- 3: $p = b_n + (x - x_k) \cdot p$
- 4: **end for**