

Computergestützte Mathematik zur Linearen Algebra

Wiederholung von Grundlagen aus der linearen Algebra

Achim Schädle

29. 11. + 6. 12. + 20. 12. 2018

$$1 \quad \mathbb{P}_m := \left\{ \sum_{i=0}^m a_i X^i \mid a_i \in \mathbb{R} \right\}$$

Addition:

$$p + q := \sum_{i=0}^m (p_i + q_i) X^i \text{ für } p := \sum_{i=0}^m p_i X^i \text{ und } q := \sum_{i=0}^m q_i X^i$$

Skalarmultiplikation:

$$\alpha \cdot p := \sum_{i=0}^m (\alpha p_i) X^i \text{ für } p := \sum_{i=0}^m p_i X^i \text{ und } \alpha \in \mathbb{R}.$$

$$2 \quad \text{Reelle } 2 \times 2 \text{ Matrizen}$$

Addition:

$$A + B := (a_{ij} + b_{ij})_{i,j=1}^2 \text{ für } A = (a_{ij})_{i,j=1}^2 \text{ und } B = (b_{ij})_{i,j=1}^2 \\ (a_{ij}, b_{ij} \in \mathbb{R})$$

Skalarmultiplikation:

$$\alpha \cdot A := (\alpha a_{ij})_{i,j=1}^2 \text{ für } A = (a_{ij})_{i,j=1}^2 \text{ und } \alpha \in \mathbb{R}.$$

Lineare (Un-)Abhängigkeit

Definition

Eine Menge von n Vektoren $M = \{u_0, \dots, u_{n-1}\} \subset V$ heißt *linear unabhängig* falls gilt

$$\sum_{j=0}^{n-1} \alpha_j u_j = 0 \Rightarrow \alpha_j = 0 \text{ für alle } j = 0, \dots, n-1$$

Definition

Ein linear unabhängiges Erzeugendensystem heißt *Basis*.

Definition

Die Anzahl der Elemente einer Basis eines Vektorraums heißt *Dimension* des Vektorraums

Welche Dimension hat der Vektorraum der symmetrischen 4×4 Matrizen?

Lineare Abbildungen

Definition

Seien U, V zwei \mathbb{R} -Vektorräume. Eine Abbildung $\alpha : U \rightarrow V$ heißt *linear*, falls

- $\alpha(u + v) = \alpha(u) + \alpha(v)$
- $\alpha(zu) = z\alpha(u)$

für alle Vektoren $u, v \in U$ und reelle z gilt.

Welche Abbildungen sind linear?

1 $\mathbb{R}^2 \rightarrow \mathbb{R}^2 : (x, y) \mapsto (-y, x)$

2 $\mathbb{R}^2 \rightarrow \mathbb{R}^2 : (x, y) \mapsto (xy, 0)$

3 $\mathbb{R}^2 \rightarrow \mathbb{R} : (x, y) \mapsto x - y$

4 $\mathbb{R}^2 \rightarrow \mathbb{R} : (x, y) \mapsto x$

Lineare Abbildungen und Matrizen: Fragestellungen

Eigenschaften von Matrizen

- $\text{kern}(A) = \{v \in V \mid Av = 0\} \subseteq V$
- $\text{bild}(A) = \{w \in W \mid \exists v \in V, Av = w\} \subseteq W$
- Basen für $\text{kern}(A)$ und $\text{bild}(A)$
- $\text{rang}(A) = \dim(\text{bild}(A))$
- Falls $n = m$, ist $A \in \text{Gl}(n)$? Existiert A^{-1} ?
- Zu $A \in \text{Gl}(n)$ löse

$$Ax = b, \quad b \in \mathbb{R}^n$$

- Zu $A \in \mathbb{R}^{m \times n}$ finde $x \in \mathbb{R}^n$, so dass

$$\|Ax - b\|_2 \rightarrow \min$$

Lösung mit Hilfe der Zeilenstufenform

Die Lösung von $Ax = b$ ist gegeben durch $x = A^{-1}b$

Ansatz: Berechne A^{-1} mit Hilfe der **spez. Zeilenstufenform**

$$\left[A \mid I \right] \longrightarrow \left[Z_A \mid E_l \cdots E_1 E_0 I \right] = \left[I \mid A^{-1} \right]$$

Berechne $x = A^{-1}b$.

Da $A^{-1} = E_l \cdots E_1 E_0$ (E_i Elementarmatrix), ist

$$A^{-1}b = E_l \cdots E_1 E_0 b.$$

Statt A^{-1} , berechne direkt spez. Zeilenstufenform von

$$\left[A \mid b \right] \longrightarrow \left[Z_A \mid E_l \cdots E_1 E_0 b \right] = \left[I \mid A^{-1}b \right]$$

Elementarmatrizen

Definition Matrizen der Form $E = I - uv^T$, $u, v \in \mathbb{R}^n$ mit $u^T v \neq 1$ heißen Elementarmatrizen.

$$uv^T = \begin{bmatrix} u_0 \\ \vdots \\ u_{n-1} \end{bmatrix} [v_0, \dots, v_{n-1}] = \begin{bmatrix} u_0 v_0 & \cdots & u_0 v_{n-1} & \vdots \\ u_{n-1} v_0 & \cdots & u_{n-1} v_{n-1} & \vdots \end{bmatrix}$$

$$v^T u = [v_0, \dots, v_{n-1}] \begin{bmatrix} u_0 \\ \vdots \\ u_{n-1} \end{bmatrix} = \sum_{i=0}^{n-1} v_i u_i$$

E ist invertierbar, $E^{-1} = I - \frac{1}{v^T u - 1} uv^T$

Spezielle Elementarmatrizen

- Multiplikation der i -ten Zeile mit $\alpha \in \mathbb{R}$ oder \mathbb{C}

$$E = I - (1 - \alpha)e_i e_i^T$$

- Addition der i -ten Zeile zur j -ten Zeile

$$E = I + e_j e_i^T$$

- Addition des α -fachen der i -ten Zeile zur j -ten Zeile

$$E = I + \alpha e_j e_i^T$$

- Vertausche i -te und j -te Zeile

$$E = I - uu^T, \quad u = e_i - e_j$$

Zeilenstufenform

Erinnerung: Jede Matrix kann durch elementare Zeilenumformungen (oder äquivalent durch Multiplikation von links mit Elementarmatrizen) auf Zeilenstufenform Z transformiert werden:

$$A \rightarrow Z = \begin{bmatrix} \otimes & \times & \dots & \dots & \dots & \dots & \dots & \times \\ 0 & 0 & \otimes & & & & & \vdots \\ \vdots & \vdots & 0 & \otimes & \times & \times & & \vdots \\ \vdots & \vdots & \vdots & 0 & 0 & 0 & \otimes & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$\otimes = \text{Pivotelemente} \neq 0$

Zeilenstufenform II

Eigenschaften der Zeilenstufenform Z von A

- Pivotpositionen sind eindeutig durch Elemente von A bestimmt
- Einträge in Z sind nicht eindeutig bestimmt
- Spalten von A , welche Pivotpositionen enthalten heißen **Basisspalten**
- $\text{Rang}(A)$ = # Basisspalten von A
= # von Null verschiedene Zeilen von Z
= # Pivotelemente

Beispiel

Aufgabe: Bestimme den Rang und identifiziere die Basisspalten von

$$A = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 2 & 4 & 2 & 2 \\ 3 & 6 & 3 & 4 \end{bmatrix}$$

Lösung: Zeilenstufenform

$$Z = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

damit: $\text{Rang } A = 2$, Pivotelemente in 1. und 4. Spalte

Basisspalten: 1. und 4. Spalte von A ($A_{:,0}, A_{:,3}$)

Spezielle Zeilenstufenform

Erinnerung: Z ist in spezieller Zeilenstufenform Z_A , wenn

- Z in Zeilenstufenform ist
- alle Pivotelemente eins sind
- alle Einträge oberhalb der Pivotelemente Null sind

$$Z_A = \begin{bmatrix} 1 & \times & 0 & 0 & \times & \times & 0 & \times \\ 0 & 0 & 1 & 0 & \times & \times & 0 & \times \\ \vdots & \vdots & 0 & 1 & \times & \times & 0 & \times \\ \vdots & \vdots & \vdots & 0 & 0 & 0 & 1 & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Z_A ist eindeutig bestimmt

Beispiel

Aufgabe: Bestimme die spezielle Zeilenstufenform Z_A , den Rang und die Basisspalten von

$$A = \begin{bmatrix} 1 & 2 & 2 & 3 & 1 \\ 2 & 4 & 4 & 6 & 2 \\ 3 & 6 & 6 & 9 & 6 \\ 1 & 2 & 4 & 5 & 3 \end{bmatrix}$$

Lösung:

$$Z_A = \begin{bmatrix} 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

damit: $\text{Rang } A = 3$, Basisspalten: $A_{:,0}, A_{:,2}, A_{:,4}$

Berechnung der Inversen einer Matrix

Erinnerung: Ist $A \in \text{GL}(n)$, also invertierbar, dann ist die spezielle Zeilenstufenform die Identität: $Z_A = I_n$

$$E^l \dots E^2 E^1 A = I \iff E^l \dots E^2 E^1 = A^{-1} = E^l \dots E^2 E^1 I$$

Gauß-Jordan-Verfahren: Transformiere die erweiterte Matrix

$$[A \mid I] \longrightarrow [I \mid A^{-1}]$$

Beispiel:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix} \longrightarrow A^{-1} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

Inverse einer Matrix

Eigenschaften der Inversen: Es seien $A, B \in \mathbb{K}^{n \times n}$ nicht singulär, dann gilt

- $(A^{-1})^{-1} = A$
- AB ist nicht singulär
- $(AB)^{-1} = B^{-1}A^{-1}$
- $(A^{-1})^T = (A^T)^{-1}$, $(A^{-1})^* = (A^*)^{-1}$,

Gauß-Elimination (Zeilenstufenform für $A \in GL(n)$)

Nach $k - 1$ Schritten Gauß-Elimination erhält man

$$A \longrightarrow A_{k-1} = \begin{bmatrix} \times & \cdots & \times & \alpha_1 & \times & \cdots & \times \\ & & & \vdots & \vdots & & \vdots \\ & & & & & & & \vdots \\ & & & \times & \alpha_{k-1} & \times & \cdots & \times \\ & & & & \alpha_k & \times & \cdots & \times \\ & & & & \vdots & \vdots & & \vdots \\ & & & & \alpha_n & \times & \cdots & \times \end{bmatrix}$$

k -ter Schritt

if $A[k,k] = 0$

Finde Index l , so dass $A[l,k] \neq 0$

Tausche Zeilen $A[k,:]$ und $A[l,:]$

Eliminiere Einträge $A[k+1:,:]$

Eliminationsschritt I

Erinnerung: Subtraktion des $l_{jk} = \alpha_j/\alpha_k$ -fachen der k -ten Zeile von der j -ten Zeile entspricht Links-Multiplikation mit

$$I - l_{jk}e_j e_k^T$$

Elimination von $A[k+1:, k] \Leftrightarrow$ Links-Multiplikation mit L_k

$$L_k = (I - l_{n-1,k}e_{n-1}e_k^T) \cdots (I - l_{k+2,k}e_{k+2}e_k^T)(I - l_{k+1,k}e_{k+1}e_k^T)$$

Wegen $\langle e_i, e_j \rangle_2 = e_j^T e_i = 0, \quad i \neq j$

$$\begin{aligned} L_k &= I - l_{n,k}e_n e_k^T - \dots - l_{k+2,k}e_{k+2}e_k^T - l_{k+1,k}e_{k+1}e_k^T \\ &= I - l_k e_k^T, \quad l_k = [0 \quad \cdots \quad 0 \quad l_{k+1,k} \quad \cdots \quad l_{n-1,k}]^T \end{aligned}$$

Eliminationsschritt II

Nach Konstruktion von L_k gilt dann

$$A_k = L_k \cdot A_{k-1} = \begin{bmatrix} \times & \cdots & \times & \alpha_0 & \times & \cdots & \times \\ & & & \vdots & \vdots & & \vdots \\ & & & & & & & \ddots \\ & & & \times & \alpha_{k-1} & \times & \cdots & \times \\ & & & & \alpha_k & \times & \cdots & \times \\ & & & & 0 & \vdots & & \vdots \\ & & & & \vdots & \vdots & & \vdots \\ & & & & 0 & \times & \cdots & \times \end{bmatrix}$$

$$= (I - \ell_k e_k^T) A_{k-1} = A_{k-1} - \ell_k e_k^T A_{k-1},$$

Beobachtung

L_k ändert die ersten $k - 1$ Zeilen von A_{k-1} nicht

Algorithmus (ohne Vertauschungen)

Mit $A_0 = A$ und $A_k = L_k A_{k-1}$ erhält man

$$L_{n-1} \cdots L_1 \cdot L_0 A = R \quad \text{obere Dreiecksmatrix}$$

Wegen $L_k^{-1} = I + \ell_k e_k^T$ ist dann

$$A = L_0^{-1} L_1^{-1} \cdots L_{n-1}^{-1} R = LR$$

$$\text{Mit } L = (I + \ell_0 e_1^T)(I + \ell_1 e_1^T) \cdots (I + \ell_{n-1} e_{n-1}^T)$$

$$= \begin{bmatrix} 1 & & & & \\ \ell_{1,0} & 1 & & & \\ \vdots & \ddots & \ddots & & \\ \ell_{n-1,0} & \cdots & \ell_{n-1,n-2} & 1 & \end{bmatrix}$$

Beispiel

Die Elemente von L und R können auf den Speicherplätzen von A gespeichert werden.

$$A = \begin{bmatrix} 2 & 2 & 2 \\ 4 & 7 & 7 \\ 6 & 18 & 22 \end{bmatrix} \longrightarrow \begin{bmatrix} 2 & 2 & 2 \\ \boxed{2} & 3 & 3 \\ 3 & 12 & 16 \end{bmatrix} \longrightarrow \begin{bmatrix} 2 & 2 & 2 \\ \boxed{2} & \boxed{3} & 3 \\ 3 & 4 & \boxed{4} \end{bmatrix}$$

Wir erhalten also die LR -Zerlegung

$$A = LR \quad \text{mit} \quad L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 2 & 2 & 2 \\ 0 & 3 & 3 \\ 0 & 0 & 4 \end{bmatrix}$$

LR-Zerlegung – Algorithmus

```
L = np.eye(n);  
  
for k in range(n-1):  
  
    for i in range(k+1, n+1):  
        L[i,k] = A[i,k] / A[k,k];  
        for j in range(k, n+1):  
            A[i,j] = A[i,j] - L[i,k]*A[k,j];  
R = A;
```

LR-Zerlegung – Algorithmus

```
L = np.eye(n);
P = np.eye(n);
for k in range(n-1):
    if A[k,k] == 0:
        Suche Index l > k, so dass not A[l,k] == 0
        Tausche Zeilen k und l von A, L und P und
        Spalten k und l von L

    for i in range(k+1, n+1):
        L[i,k] = A[i,k] / A[k,k];
        for j in range(k, n+1):
            A[i,j] = A[i,j] - L[i,k]*A[k,j];
R = A;
```


LR-Zerlegung – Platzsparende Variante

```
p = range(n);
for k in range(n - 1):
    if A[k,k] == 0:
        Suche Index l > k, so dass not A[l,k] == 0
        Tausche Zeilen k und l von A
        vertausche Elemente k und l von p

    for i in range(k+1, n+1):
        A[i,k] = A[i,k] / A[k,k];
        A[i,k+1:] = A[i,k+1:] - A[i,k] * A[k, k+1:];
```

Dann:

```
R = np.triu(A);
L = np.eye(n) + np.tril(A,-1);
P = Permutationsmatrix, die p[k] auf k abbildet, k = 0,
```

- j -Schleife vektorisiert
- A zum Speichern von L und R verwendet

Aufwand

$$\begin{aligned}\sum_{k=0}^{n-2} \sum_{i=k+1}^n (n-k+1) &= \sum_{k=1}^{n-1} (n-k)(n-k+1) \\ &= \sum_{j=1}^{n-1} j(j+1) \\ &= \frac{1}{6}(n-1)n(2n-1) + \frac{1}{2}n(n-1) \\ &= \frac{1}{3}n^3 - \frac{1}{3}n \\ &= \frac{1}{3}n^3 + \mathcal{O}(n^2)\end{aligned}$$

Erinnerung: $\sum_{j=1}^m j^2 = \frac{1}{6}m(m+1)(2m+1)$

Satz (LR-Zerlegung)

Sei $A \in \mathbb{K}^{n \times n}$ so, dass bei der Reduktion auf Zeilenstufenform ohne Zeilenvertauschung kein Pivotelement 0 auftritt. Dann gibt es eine Zerlegung $A = LR$ mit

- 1 L ist untere und R ist obere Dreiecksmatrix
- 2 $\ell_{i,i} = 1$ und $r_{i,i} \neq 0$ für $i = 1, 2, \dots, n$
- 3 L und R sind damit eindeutig bestimmt.

Anwendungen der LR-Zerlegung

Lösung linearer Gleichungssysteme

Sei $A = LR$ die LR-Zerlegung

$$Ax = b$$

$$\iff L(Rx) = b$$

$$\iff Ly = b, \quad Rx = y$$

- L und R haben Dreiecksform
- $Ly = b \rightarrow$ Vorwärtssubstitution
- $Rx = y \rightarrow$ Rückwärtssubstitution

Vorwärtssubstitution

$$Ly = b \iff \begin{bmatrix} 1 & & & \\ \ell_{2,1} & 1 & & \\ \vdots & \ddots & \ddots & \\ \ell_{n,1} & \cdots & \ell_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Lösung

```
for i in range(n):  
    y[i] = b[i];  
    for j in range(i):  
        y[i] -= L[i,j]*y[j];
```

Aufwand $\sum_{k=2}^n (k-1) = \frac{1}{2}n(n-1) = \frac{1}{2}n^2 + O(n)$

Rückwärtssubstitution

$$Rx = y \iff \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ & \ddots & \ddots & \vdots \\ & & r_{n-1,n-1} & r_{n-1,n} \\ & & & r_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Lösung

```
for i in range(n-1, -1, -1): # Rückwärts von n-1, ..., 0
    x[i] = y[i];
    for j in range(i+1:n+1):
        x[i] -= R[i,j]*x[j];
    x[i] = x[i]/R[i,i];
```

Aufwand $\sum_{k=1}^{n-1} (n-k) = \sum_{k=1}^{n-1} k = \frac{1}{2}n^2 + O(n)$

Bemerkungen zur LR-Zerlegung

- Ist die LR-Zerlegung von A schon berechnet und möchte man ein weiteres Gleichungssystem $Ax = c$ lösen, so kann man die **teure** LR-Zerlegung ($\sim \frac{1}{3}n^3$ Operationen) wiederverwenden und muss nur die **billigen** Vorwärts-/Rückwärtseliminationen (je $\sim \frac{1}{2}n^2$ Operationen) neu rechnen.
- Wendet man das Gauß-Verfahren direkt auf die erweiterte Matrix $[A \mid b]$ an, so ergibt sich $[R \mid L^{-1}b]$, die Vorwärtselimination kann also einfach mitgerechnet werden.

Darstellungsmatrix

Definition

Seien U, V zwei \mathbb{K} -Vektorräume. Seien $\{u_1, \dots, u_n\} \subseteq U$ und $\{v_1, \dots, v_m\} \subseteq V$ Basen. Sei $\varphi : U \rightarrow V$ linear. Seien $\varphi(u_j) = \sum_{i=1}^m a_{ij} v_i$, $1 \leq j \leq n$. Dann heißt

$$A = (a_{ij})_{i,j=1}^{m,n} \in \mathbb{K}^{m \times n}$$

eine **Darstellungsmatrix** der linearen Abbildung φ .

Bemerkung

- Die Koeffizienten a_{ij} sind durch die Wahl der Basen von U und V eindeutig bestimmt.
- Seien $u = \sum_{j=1}^n \mu_j u_j$ und $\varphi(u) =: v = \sum_{i=1}^m \nu_i v_i$. Definiere $\mu := (\mu_j)_{j=1}^n$ und $\nu := (\nu_i)_{i=1}^m$, so gilt

$$\nu = A\mu, \quad \text{also} \quad \nu_i = \sum_{j=1}^n a_{ij} \mu_j, \quad 1 \leq j \leq n.$$

Bilinearformen

Sei V ein \mathbb{R} -Vektorraum.

Definition

Eine Abbildung $a : V \times V \rightarrow \mathbb{R}$ heißt **Bilinearform auf V** , falls

- $a(u, v + w) = a(u, v) + a(u, w)$
- $a(u + w, v) = a(u, v) + a(w, v)$
- $a(\alpha u, v) = \alpha a(u, v)$
- $a(u, \alpha v) = \alpha a(u, v)$

für alle Vektoren u, v, w und $\alpha \in \mathbb{R}$ gilt.

D.h. die Abbildungen $a(\cdot, v)$ und $a(u, \cdot)$ sind beide linear.

Definition

a heißt **positiv (semi-) definit**, falls $\mathbb{R} \ni a(u, u) > 0$ bzw. ≥ 0 für alle $0 \neq u \in V$.

Sesquilinearformen

Sei V ein \mathbb{C} -Vektorraum.

Definition

Eine Abbildung $a : V \times V \rightarrow \mathbb{C}$ heißt **Sesquilinearform auf V** , falls

- $a(u, v + w) = a(u, v) + a(u, w)$
- $a(u + w, v) = a(u, v) + a(w, v)$
- $a(\alpha u, v) = \alpha a(u, v)$
- $a(u, \alpha v) = \bar{\alpha} a(u, v)$

für alle Vektoren u, v, w und $\alpha \in \mathbb{C}$ gilt.

D.h. $a(\cdot, v)$ ist linear und $a(u, \cdot)$ ist **semilinear**.

Definition

a heißt **positiv (semi-) definit**, falls $\mathbb{R} \ni a(u, u) > 0$ bzw. ≥ 0 für alle $0 \neq u \in V$.

Skalarprodukt / inneres Produkt

Sei V ein \mathbb{K} -Vektorraum.

Definition

Eine Bi-/Sesquilinearform a auf V heißt **symmetrisch/hermitesch**, falls $a(u, v) = \overline{a(v, u)}$ für alle $u, v \in V$.

Definition

Eine symmetrische/hermitesche, positiv definite Bi-/Sesquilinearform ist ein **Skalarprodukt**.

Beispiel:

Für $u, v \in \mathbb{R}^n$ ist

$$(u, v) \mapsto \sum_{k=1}^n u_k v_k = v^T u =: \langle u, v \rangle$$

ein Skalarprodukt (das **euklidische Skalarprodukt** des \mathbb{R}^n).

Beispiele Skalarprodukte

- 1 $(A, B) \mapsto \sum_{i,j=1}^n a_{ij}b_{ij}$ ist ein Skalarprodukt auf dem Vektorraum der reellen $n \times n$ Matrizen.
- 2 $(p, q) \mapsto \int_0^1 p(t)q(t)dt$ ist ein Skalarprodukt auf \mathbb{P}_m mit reellen Koeffizienten.

Unitäre und Orthogonale Abbildungen

Definition

Sei A eine lineare Abbildung auf einem \mathbb{R} -Vektorraum mit Skalarprodukt $(V, \langle \cdot, \cdot \rangle)$. Falls $\|Ax\| = \|x\|$ für alle $x \in V$ gilt, so heißt A orthogonal.

Definition

Sei A eine lineare Abbildung auf einem \mathbb{C} -Vektorraum mit Skalarprodukt $(V, \langle \cdot, \cdot \rangle)$. Falls $\|Ax\| = \|x\|$ für alle $x \in V$ gilt, so heißt A unitär.

hierbei ist $\|x\| := \langle x, x \rangle^{1/2}$ die durch das Skalarprodukt gegebene Norm.

Beispiele

- ① In \mathbb{R}^2 ist die Drehung gegeben durch die Matrix

$$\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

orthogonal.

- ② In \mathbb{C}^3 ist die Spiegelung an der (x, y) -Ebene

$$(x, y, z) \mapsto (x, y, -z)$$

unitär.

- ③ In \mathbb{R}^3 ist die Projektion auf die (x, y) -Ebene

$$(x, y, z) \mapsto (x, y, 0)$$

nicht orthogonal.

Orthogonal- und Orthonormalbasen

Definition

Sei V ein \mathbb{K} -Vektorraum mit Skalarprodukt $\langle \cdot, \cdot \rangle_V$ (reell oder komplex). Eine Menge $\{v_1, \dots, v_n\} \subseteq V$ heißt **orthogonal**, falls

$$\langle v_i, v_j \rangle_V = 0, \quad \text{falls } i \neq j.$$

Eine orthogonale Menge $\{v_1, \dots, v_n\} \subseteq V$ heißt **orthonormal**, falls

$$\langle v_i, v_i \rangle_V = 1, \quad \text{für } 1 \leq i \leq n.$$

Entsprechende Basen heißen **Orthogonalbasis** bzw. **Orthonormalbasis**.

Ab jetzt für heute wir versehen \mathbb{R}^n mit dem euklidischen Skalarprodukt $\langle u, v \rangle = v^T u$.

Beispiel

Seien $\{v_1, \dots, v_n\} \subseteq \mathbb{R}^m$ *orthonormale* Vektoren, so ist die Matrix

$$Q = [v_1 \mid \cdots \mid v_n]$$

orthogonal.

Umgekehrt sind die Spalten einer *orthogonalen* Matrix *orthonormale* Vektoren.

Gram-Schmidt-Orthogonalisierung

Erinnerung:

Linear unabhängige Vektoren a_1, \dots, a_n in \mathbb{R}^m können zu q_1, \dots, q_n orthonormalisiert werden:

$$\langle a_1, \dots, a_k \rangle = \langle q_1, \dots, q_k \rangle, \quad k = 1, \dots, n$$

mit

$$q_j^T q_k = \begin{cases} 0, & k \neq j \\ 1, & k = j \end{cases}.$$

Gram-Schmidt-Orthogonalisierung: Für $k = 1, \dots, n$

$$\begin{aligned} \tilde{q}_k &= a_k - \sum_{j=1}^{k-1} r_{jk} q_j, & r_{jk} &= q_j^T a_k \\ q_k &= \tilde{q}_k / r_{kk}, & r_{kk} &= \|\tilde{q}_k\| \end{aligned}$$

QR-Zerlegung für $A \in \mathbb{R}^{m \times n}$

Gram-Schmidt-Orthogonalisierung: für $k = 1, \dots, n$

$$\tilde{q}_k = a_k - \sum_{j=1}^{k-1} r_{jk} q_j, \quad q_k = \tilde{q}_k / r_{kk}$$

$$\iff a_k = \sum_{j=1}^k r_{jk} q_j = [q_1 \mid q_2 \mid \dots \mid q_k] \begin{bmatrix} r_{1k} \\ \vdots \\ r_{kk} \end{bmatrix}$$

$$\iff [a_1 \mid a_2 \mid \dots \mid a_n] = [q_1 \mid q_2 \mid \dots \mid q_n] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}$$

reduzierte und volle QR-Zerlegung

Reduzierte QR-Zerlegung

$$A = Q \cdot R$$

$$Q^T Q = \text{Id}_n$$

Volle QR-Zerlegung

$$A = \hat{Q} \cdot \hat{R}$$

$$\hat{Q}^T \hat{Q} = \hat{Q} \hat{Q}^T = \text{Id}_m$$

Reduzierte QR-Zerlegung

Definition

Sei $A \in \mathbb{R}^{m \times n}$ mit $\text{Rang}(A) = n < m$ gegeben. Dann heißt

- $A = QR$ mit $Q \in \mathbb{R}^{m,n}$ (spalten-) orthogonal und oberer Dreiecksmatrix $R \in \mathbb{R}^{n,n}$ heißt **reduzierte QR-Zerlegung**.
- $A = \hat{Q}\hat{R}$ mit $\hat{Q} = [Q \ Q_{\perp}] \in \mathbb{R}^{m,m}$ orthogonal und oberer Dreiecksmatrix $\hat{R} = \begin{bmatrix} R \\ 0_{m-n,n} \end{bmatrix} \in \mathbb{R}^{m,n}$ heißt **volle QR-Zerlegung** von A .

Satz

Die reduzierte QR-Zerlegung von A ist eindeutig, wenn man $r_{ii} > 0$ voraussetzt.

Modifiziertes Gram-Schmidt-Verfahren

Klassische Gram-Schmidt-Orthogonalisierung:

$$\begin{aligned}\tilde{q}_k &= a_k - \sum_{j=1}^{k-1} r_{jk} q_j, & r_{jk} &= q_j^T a_k \\ q_k &= \tilde{q}_k / r_{kk}, & r_{kk} &= \|\tilde{q}_k\|\end{aligned}$$

Interpretation (falls die Summe in der Reihenfolge $1, 2, \dots$ berechnet wird):

- orthogonalisiere a_k (gleichzeitig) gegen q_1, \dots, q_{k-1}

Modifikation:

- orthogonalisiere \tilde{q}_k nacheinander gegen q_1, \dots, q_{k-1}

Modifiziertes Gram-Schmidt-Verfahren

- orthogonalisiere a_j zunächst gegen q_1 :

$$\tilde{q}_{1k} = a_k - r_{1k}q_1, \quad r_{1k} = q_1^T a_k$$

- anschließend orthogonalisiere \tilde{q}_{1k} gegen q_2 :

$$\tilde{q}_{2k} = \tilde{q}_{1k} - r_{2k}q_2, \quad r_{2k} = q_2^T \tilde{q}_{1k}$$

- anschließend orthogonalisiere \tilde{q}_{2k} gegen q_3 :

$$\tilde{q}_{3k} = \tilde{q}_{2k} - r_{3k}q_3, \quad r_{3k} = q_3^T \tilde{q}_{2k}$$

- USW.

Vor- und Nachteile

- Orthogonalisierungen *müssen* nacheinander durchgeführt werden (d.h. keine Parallelisierung möglich)
- in exakter Arithmetik sind beide Algorithmen äquivalent, denn

$$\begin{aligned}r_{jk} &= q_j^T a_k \\ &= q_j^T \left(a_k - \sum_{i=1}^{j-1} r_{ik} q_i \right)\end{aligned}$$

da $q_j^T q_i = 0$ für $j \neq i$.

- in Gleitkommaarithmetik ist der modifizierte Gram-Schmidt-Algorithmus stabiler

Klassisches und Modifiziertes Gram Schmidt Verfahren

Gegeben Basisvektoren $[a_1, \dots, a_n]$

for $k = 1, \dots, n$ **do**

$$q_k = a_k$$

if $k \neq 1$ **then**

$$r_{j,k} = q_j^T q_k, \text{ für } j = 1, \dots, k-1$$

$$q_k = q_k - \sum_{j=1}^{k-1} q_j r_{j,k}$$

end if

$$r_{k,k} = \|q_k\|$$

$$q_k = q_k / r_{k,k}$$

end for

Erhalte Basisvektoren q_1, \dots, q_n

und Matrix R

Gegeben Basisvektoren $[a_1, \dots, a_n]$

for $k = 1, \dots, n$ **do**

$$q_k = a_k$$

for $i = 1..k - 1$ **do**

$$r_{i,k} = q_i^T q_k,$$

$$q_k = q_k - q_i r_{i,k}$$

end for

$$r_{k,k} = \|q_k\|$$

$$q_k = q_k / r_{k,k}$$

end for

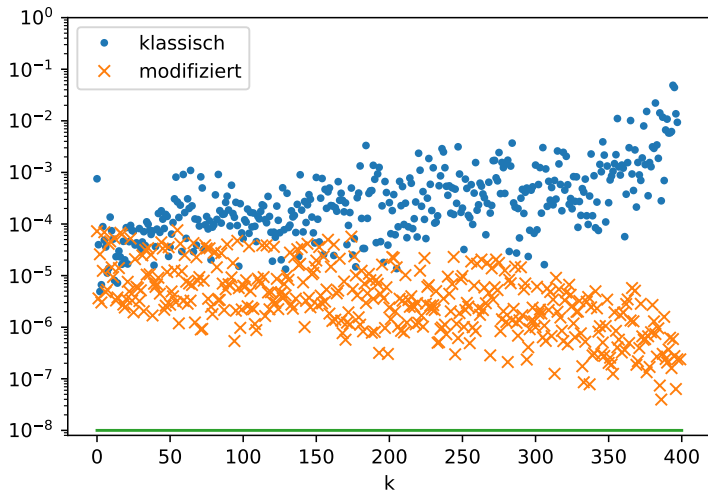
Erhalte Basisvektoren q_1, \dots, q_n

und Matrix R

Stabilitätsvergleich

Klassisches und modifiziertes Gram-Schmidt-Verfahren

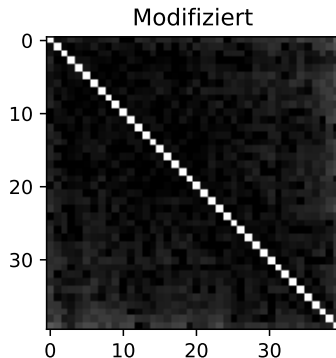
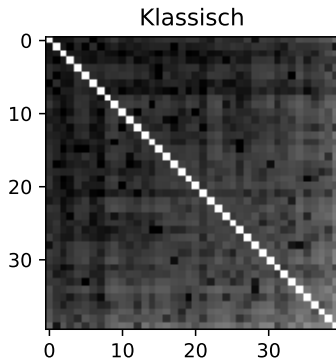
Orthogonalität von Q : $\max_{j>k} |q_k^T q_j|$



Stabilitätsvergleich

Klassisches und modifiziertes Gram-Schmidt-Verfahren

Einträge von $\log_{10}(|Q^T \cdot Q|)$. Weiß $\hat{=}$ 0 = $\log_{10}(1)$, Schwarz $\hat{=}$ $-8 = \log_{10}(10^{-8}) = \log_{10}(\text{eps})$,



Lösung linearer Gleichungssysteme mit QR-Zerlegung

Für $A \in \mathbb{R}^{n,n}$ nicht singulär ist

$$Ax = b \iff QRx = b \iff Rx = Q^T b$$

Wie löst man mit Q ? Tipp: $Q^T Q = \text{Id}$

I.A. teurer aber (mit mod. Gram-Schmidt oder Householder-Transformationen) stabiler als Gauß-Elimination.