

Mathematik für InformatikerInnen 2

Frank-Olaf Schreyer

Universität des Saarlandes, SS 2020

Einleitung

Thema sind Vektorräume und lineare Algebra. Das Konzept der Vektorräume ist die Struktur, die am häufigsten in der Mathematik und deren Anwendungen auftritt. In der Informatik werden sie zum Beispiel angewendet bei folgendem:

- ▶ Google page rank
- ▶ Fehlerkorrigierende Codes, z.B. TCP/IP-Protokoll
- ▶ Kryptographie, Vektorräume über endlichen Körpern + algebraische Geometrie (elliptische Kurven usw.)
- ▶ Raytracing
- ▶ Datenkompression, JPEG
- ▶ ML, z.B. Vektorquantisierung
- ▶ Schnelle Multiplikation, diskrete Fourier Transformation.
- ▶ Komplexitätstheorie, Tensor Rank

Gliederung der Vorlesung

1. \mathbb{R}^n und der Gauß-Algorithmus
2. Abstrakte Vektorräume
3. Matrizen und lineare Abbildungen
4. Gruppen und Symmetrie
5. Determinanten
6. Eigenwerte und das Charakteristische Polynom
7. Skalarprodukte und das Gram-Schmidt Verfahren
8. Fourierreihen – noch einmal
9. Hauptachsentransformation und Quadriken
10. Singulärwertzerlegung

Ziel der heutigen Vorlesung

Wir wollen den Gauß-Algorithmus zum Lösen von linearen Gleichungssystemen kennenlernen.

1. Beispiel eines 3×3 Gleichungssystem
2. Zeilenstufenform von Matrizen und elementare Zeilenumformungen
3. Gauß-Algorithmus
4. Matrizenprodukte
5. Algorithmus zum Lösen linearer Gleichungssysteme
6. Pivotisierung

Lineare Gleichungssysteme

$$x_1 + 3x_2 - 4x_3 = 1$$

$$2x_1 + x_2 - 2x_3 = -2$$

$$x_1 + 2x_2 - 2x_3 = 3$$

Eine Strategie, wie man ein lineares Gleichungssystem löst, sollte aus der Schule bekannt sein.

Lineare Gleichungssysteme

$$x_1 + 3x_2 - 4x_3 = 1$$

$$-5x_2 + 6x_3 = -4$$

$$-x_2 + 2x_3 = 2$$

Wir eliminieren als nächstes x_2 . Um Brüche zu vermeiden, vertauschen wir zunächst die Gleichungen II und III.

$$x_1 + 3x_2 - 4x_3 = 1$$

$$-x_2 + 2x_3 = 2$$

$$-4x_3 = -14$$

Nun können wir rückwärts lösen.

Die gleiche Strategie lässt sich auf ein Gleichungssystem von m linearen Gleichungen in n Variablen anwenden

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned}$$

Die Unbestimmten sind $x_1 \dots x_n$. Die Koeffizienten sind in einer $m \times n$ -Matrix organisiert

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \in \mathbb{R}^{m \times n}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m,$$

und die rechte Seite ist ein Vektor $b \in \mathbb{R}^m$.

In gewisser Weise ist es überflüssig, die Variablen in unseren Umformungen mit hinzuschreiben, da welche Variable wo auftaucht schon durch die Position bestimmt ist. In unserem Beispiel reicht es, alle Koeffizienten hinzuschreiben. Statt

$$\begin{array}{l} x_1 + 3x_2 - 4x_3 = 1 \\ 2x_1 + x_2 - 2x_3 = -2 \\ x_1 + 2x_2 - 2x_3 = 3 \end{array} \quad \text{notieren wir} \quad \left(\begin{array}{ccc|c} 1 & 3 & -4 & 1 \\ 2 & 1 & -2 & -2 \\ 1 & 2 & -2 & 3 \end{array} \right)$$

und protokollieren die Umformungen

$$\rightsquigarrow \left(\begin{array}{ccc|c} 1 & 3 & -4 & 1 \\ 0 & -5 & 6 & -4 \\ 0 & -1 & 2 & 2 \end{array} \right) \rightsquigarrow \left(\begin{array}{ccc|c} 1 & 3 & -4 & 1 \\ 0 & -1 & 2 & 2 \\ 0 & 0 & -4 & -14 \end{array} \right)$$

Der Gauß-Algorithmus ist die Verallgemeinerung dieser Strategie.

Definition

Eine $m \times n$ Matrix $A = (a_{ij})$ ist in **Zeilenstufenform**, falls es eine natürliche Zahl r mit $0 \leq r \leq m$ und Spaltenindizes $1 \leq j_1 < j_2 < \dots < j_r \leq n$ gibt, so dass folgendes gilt:

$$\begin{cases} a_{kj_k} \neq 0 & \text{für } k = 1, \dots, r, \\ a_{kj} = 0 & \text{für } j < j_k \text{ und } k = 1, \dots, r, \\ a_{ij} = 0 & \text{für alle } i, j \text{ mit } i > r. \end{cases}$$

Beispiel: Die 4×7 Matrix

$$\begin{pmatrix} 0 & 2 & 3 & 5 & 7 & 11 & 13 \\ 0 & 0 & 1 & 4 & 9 & 16 & 25 \\ 0 & 0 & 0 & 0 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

ist in Zeilenstufenform mit $r = 3$ Stufen und Spalten $j_1 = 2, j_2 = 3$ und $j_3 = 6$.

Definition

Eine **elementare Zeilenoperation** auf einer $m \times n$ Matrix $A = (a_{ij})$ ist eine Operation vom folgenden Typ:

- I) $M_i(\lambda)$: Multiplikation der i -ten Zeile mit einem $\lambda \neq 0$, d.h. wir ersetzen $a_i = (a_{i1}, \dots, a_{in})$ durch

$$\lambda a_i = (\lambda a_{i1}, \dots, \lambda a_{in}),$$

- II) $E_{(i,j)}$: Addition der i -ten Zeile zur j -ten Zeile, d.h. wir ersetzen $a_j = (a_{j1}, \dots, a_{jn})$ durch

$$a_j + a_i = (a_{j1} + a_{i1}, \dots, a_{jn} + a_{in}),$$

- III) $E_{(i,j)}(\lambda)$: Addition des λ -fachen der i -ten Zeile zur j -ten Zeile für ein $\lambda \neq 0$, d.h. wir ersetzen $a_j = (a_{j1}, \dots, a_{jn})$ durch

$$a_j + \lambda a_i = (a_{j1} + \lambda a_{i1}, \dots, a_{jn} + \lambda a_{in}),$$

- IV) $P_{(i,j)}$: Die Permutation der i -ten mit der j -ten Zeile, d.h. wir vertauschen $a_i = (a_{i1}, \dots, a_{in})$ und $a_j = (a_{j1}, \dots, a_{jn})$,

Bem.: Operationen vom Typ III und VI kann man auch erreichen, indem man Operationen vom Typ I and II wiederholt anwendet.

Beweis. III)

$$A = \begin{pmatrix} \vdots \\ a_i \\ \vdots \\ a_j \\ \vdots \end{pmatrix} \xrightarrow{I} \begin{pmatrix} \vdots \\ \lambda a_i \\ \vdots \\ a_j \\ \vdots \end{pmatrix} \xrightarrow{II} \begin{pmatrix} \vdots \\ \lambda a_i \\ \vdots \\ a_j + \lambda a_i \\ \vdots \end{pmatrix} \xrightarrow{I} \begin{pmatrix} \vdots \\ a_i \\ \vdots \\ a_j + \lambda a_i \\ \vdots \end{pmatrix} .$$

IV)

$$A = \begin{pmatrix} \vdots \\ a_i \\ \vdots \\ a_j \\ \vdots \end{pmatrix} \xrightarrow{III} \begin{pmatrix} \vdots \\ a_i \\ \vdots \\ a_j - a_i \\ \vdots \end{pmatrix} \xrightarrow{II} \begin{pmatrix} \vdots \\ a_j \\ \vdots \\ a_j - a_i \\ \vdots \end{pmatrix} \xrightarrow{III} \begin{pmatrix} \vdots \\ a_j \\ \vdots \\ -a_i \\ \vdots \end{pmatrix} \xrightarrow{I} \begin{pmatrix} \vdots \\ a_j \\ \vdots \\ a_i \\ \vdots \end{pmatrix} .$$

Theorem

Zu einer $m \times n$ Matrix $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ gibt es eine (endliche) Folge elementarer Zeilenoperationen vom Typ III und IV, welche A in eine Matrix \bar{A} in Zeilenstufenform überführt.

Beweis.

Algorithmus.[Gauß] **Input.** Eine $m \times n$ Matrix $A = (a_{ij})$.

Output. Eine $m \times n$ Matrix $\bar{A} = (\bar{a}_{ij})$ in Zeilenstufenform.

0. **if** $m = 1$ **then return** A . Falls $A \neq 0$ gilt, dann ist die Zahl der Stufen $r = 1$ und $j_1 = \min\{j \mid a_{1j} \neq 0\}$, anderenfalls ist $r = 0$.
1. **if** $a_{11} \neq 0$ **then** $j_1 = 1$. **for** i **from** 2 **to** m wende $E_{(1,i)}(\frac{-a_{i1}}{a_{11}})$ an. Das Resultat ist eine Matrix der Gestalt

$$A' = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a'_{22} & \dots & a'_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & a'_{m2} & \dots & a'_{mn} \end{pmatrix}.$$

if $n = 1$ **then return** A' **else** ruf den Algorithmus mit der $(m-1) \times (n-1)$ Matrix $A' = (a'_{ij})_{i,j \geq 2}$ rekursiv auf, um $\bar{A}' = (\bar{a}'_{ij})_{i,j \geq 2}$ in Zeilenstufenform zu erhalten mit Stufen in Spalten $2 \leq j_2 < \dots < j_r \leq m$ und **return**

$$\bar{A} := \begin{pmatrix} a_{11} & * \\ 0 & \bar{A}' \end{pmatrix}.$$

2. **if** $a_{11} = 0$ aber die erste Spalte ist nicht identisch Null, etwa $a_{i1} \neq 0$ **then** wende die elementare Operation $P_{(1,i)}$ an, um eine Matrix A' mit $a'_{11} \neq 0$ zu erhalten. Ruf den Algorithmus mit A' auf, um \bar{A} in Spaltenstufenform zu erhalten und **return** \bar{A} .
3. Die erste Spalte ist identisch Null, d.h. alle $a_{i1} = 0$: **if** $n = 1$ **then return** die $m \times 1$ Nullmatrix **else** ruf den Algorithmus rekursiv mit der $m \times (n - 1)$ Matrix $A' = (a_{ij})_{j \geq 2}$ auf, um eine Matrix $\bar{A}' = (\bar{a}'_{ij})_{j \geq 2}$ in Zeilenstufenform zu erhalten mit Stufen in Spalten $2 \leq j_1 < \dots < j_r \leq m$. Dann **return**

$$\bar{A} = \left(0 \quad \bar{A}' \right) \quad \square$$

Der Satz folgt, da wir die elementaren Zeilenoperationen der Matrix in der Rekursion als elementare Zeilenoperation der ursprünglichen $m \times n$ Matrix auffassen können wegen der Nullen in der ersten Spalte. □

Lineare Gleichungssysteme in Matrixschreibweise

Wenn wir

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

als $n \times 1$ bzw $m \times 1$ Matrizen auffassen, können wir unser System von m linearen Gleichungen in n Variablen wie folgt schreiben:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix},$$

oder noch kürzer in der Form

$$A \cdot x = b$$

wobei $A = (a_{ij})$ die $m \times n$ Matrix der Koeffizienten ist.

Algorithmus zum Lösen linearer Gleichungssysteme

Input. Eine $m \times n$ Matrix $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ und ein Vektor $b = (b_j) \in \mathbb{R}^m$.

Output. Eine Parametrisierung der Lösungsmenge

$$\{x \in \mathbb{R}^n \mid A \cdot x = b\}$$

des linearen Gleichungssystems $Ax = b$.

1. Bilden die $m \times (n + 1)$ erweiterte Matrix

$$(A|b) = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{array} \right)$$

2. Rufen den Gauß-Algorithmus auf, um $(A|b)$ in eine Matrix $(\bar{A}|\bar{b})$ in Zeilenstufenform zu überführen.

3. Wenn die Matrix $(\bar{A}|\bar{b})$ genau r Stufen in Spalten $j_1 < \dots < j_r$ hat, dann gilt: Ist $j_r = n + 1$ dann ist die Lösungsmenge leer, da $b_r \neq 0$ und die Gleichung $0 = b_r$ kein Lösung hat.
4. Ist $j_r \leq n$ dann wird die Lösungsmenge durch $n - r$ Parametern t_1, \dots, t_{n-r} parametrisiert: Wir betrachten das Komplement $\{1, \dots, n\} \setminus \{j_1, \dots, j_r\} = \{\ell_1, \dots, \ell_{n-r}\}$ und setzen $x_{\ell_i} = t_i$.
5. **for k from r to 1** berechnen wir x_{j_k} aus der k -ten Gleichung von $\bar{A}x = \bar{b}$ unter Verwendung der schon berechneten x_j für $j > j_k$:

$$x_{j_k} = \frac{1}{\bar{a}_{kj_k}} (\bar{b}_k - \sum_{j=j_{k+1}}^n \bar{a}_{kj} x_j).$$

Beispiel. Angenommen

$$(\overline{A}|\overline{b}) = \left(\begin{array}{cccccc|c} 0 & 2 & 3 & 5 & 7 & 11 & 13 \\ 0 & 0 & 1 & 4 & 9 & 16 & 25 \\ 0 & 0 & 0 & 0 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

ist das Resultat des Gauß-Algorithmus angewendet auf die erweiterte Matrix $(A|b)$ eines linearen Gleichungssystems aus 4 Gleichungen in 6 Unbekannten. Da $j_3 = 6 < 7$ gilt, ist die Lösungsmenge nicht leer.

$$(\overline{A}|\overline{b}) = \left(\begin{array}{cccccc|c} 0 & 2 & 3 & 5 & 7 & 11 & 13 \\ 0 & 0 & 1 & 4 & 9 & 16 & 25 \\ 0 & 0 & 0 & 0 & 0 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Wir setzen $x_1 = t_1$, $x_4 = t_2$ und $x_5 = t_3$. Anschließend berechnen wir $x_6 = 2$ aus der dritten Gleichung,

$$x_3 = \qquad \qquad \qquad = -7 - 4t_2 - 9t_3$$

aus der zweiten und schließlich

$$x_2 = \qquad \qquad \qquad = 6 + \frac{7}{2}t_2 + 10t_3$$

aus der ersten Gleichung. Die Lösungsmenge ist also

$$\left\{ \begin{pmatrix} t_1 \\ 6 + \frac{7}{2}t_2 + 10t_3 \\ -7 - 4t_2 - 9t_3 \\ t_2 \\ t_3 \\ 2 \end{pmatrix} \mid t_1, t_2, t_3 \in \mathbb{R} \right\}$$

Beweis der Korrektheit des Algorithmus.

Da elementare Zeilenoperationen reversibel sind, haben die Gleichungssysteme $Ax = b$ und $\bar{A}x = \bar{b}$ die gleiche Lösungsmenge. Das Resultat folgt: x_{j_k} ist durch die k -te Gleichung von $\bar{A}x = \bar{b}$ und durch die Werte für die x_j für $j > j_k$ eindeutig bestimmt, da $\bar{a}_{kj_k} \neq 0$ ist. □

Beispiel.

$$\begin{array}{rcl} x_1 + x_2 & + x_3 + x_4 & = 1 \\ x_1 - x_2 & + x_3 - x_4 & = 2 \\ x_1 + 2x_2 & + 4x_3 + 8x_4 & = 3 \\ x_1 - 2x_2 & + 4x_3 - 8x_4 & = 4 \end{array} \quad \begin{array}{l} \text{mit der} \\ \text{erweiterten} \\ \text{Matrix} \end{array} \quad \left(\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 2 \\ 1 & 2 & 4 & 8 & 3 \\ 1 & -2 & 4 & -8 & 4 \end{array} \right)$$

Beispiel.

Bemerkung. Typischerweise, etwa für zufällig gewählte Gleichungssysteme, ist die Anzahl der Stufen $r = \min(m, n + 1)$ und die Stufen haben Zeilenindizes $j_k = k$ für $k = 1, \dots, r$. Folglich hat man im Fall $m > n$ in der Regel keine Lösung, im Fall $m = n$ meistens genau eine Lösung und im Fall $m < n$ in der Regel eine $(n - m)$ -dimensionale Lösungsmenge.

Wir werden diese Fragestellung später erneut aufgreifen.

Pivotelemente

1. Es kann nützlich sein, $M_1(\frac{1}{a_{11}})$ im Fall 1) anzuwenden, um $a'_{11} = 1$ zu erreichen bevor wir die Spalte ausräumen.
2. Im Fall von Matrizen $A \in \mathbb{Q}^{m \times n}$ can es nützlich sein eine Permutation P_{1j} anzuwenden, welche das Element, dessen Zähler und Nenner zusammen die wenigsten Ziffern hat, in die erste Position zu bewegen. Das kann helfen, große Brüche zu vermeiden.
3. Wenn man mit Fließkommazahlen arbeitet, z.B. \mathbb{R}_{53} mit 53 Binärziffern, verwendet man **Pivotisierung**, um Rundungsfehler so gut es geht zu vermeiden. Man bewegt den Eintrag der ersten Spalte, welcher den größten Absolutbetrag hat, in die (1, 1) Position.
(Wenn man auch noch eine Permutation der Unbestimmten also der Spalten vornimmt, kann man den insgesamt größten Eintrag der Matrix in die (1, 1) Position bewegen.)

Beispiel. Wir bearbeiten das folgende lineares Gleichungssystem mit einer Rechengenauigkeit von 3 Dezimalstellen.

$$\begin{pmatrix} 1.00 \cdot 10^{-4} & 1.00 \\ 1.00 & 1.00 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1.00 \\ 2.00 \end{pmatrix}$$