Lineare Algebra Woche 3

Gleichungssysteme, Kern/Bild und Codierungstheorie

Themen: Gauß-Algorithmus, Rang, Kern/Bild, Basiswechsel, Lineare Codes, Hamming-Codes

Übersicht

Teil I: Gauß-Algorithmus und Rang
- Lineare Gleichungssysteme
- Gauß-Elimination
- Rangbestimmung

Teil II: Kern und Bild
- Nullraum einer Matrix
- Spaltenraum und Dimension
- Basisbestimmung

Teil III: Basiswechsel
- Darstellungsmatrizen
- Koordinatentransformation

Teil IV: Codierungstheorie
- Motivation und Grundbegriffe
- Lineare Codes
- Hamming-Distanz

Teil V: Hamming-Codes
- Generatormatrix
- Prüfmatrix
- Fehlerkorrektur

Anhänge
- Übungsaufgaben
- Praktische Beispiele

Diese Woche: Von der Theorie zur praktischen Anwendung in der digitalen Kommunikation

Teil I: Gauß-Algorithmus und Rang

Lineare Gleichungssysteme

Allgemeine Form

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

Matrixform

$$A\vec{x} = \vec{b}$$

wobei $A \in \mathbb{R}^{m \times n}$, $\vec{x} \in \mathbb{R}^n$, $\vec{b} \in \mathbb{R}^m$

Erweiterte Koeffizientenmatrix

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

Lösungsmöglichkeiten:

- Keine Lösung (inkonsistent)
- Eindeutige Lösung
- Unendlich viele Lösungen

Lineare Gleichungssysteme sind fundamental für alle Bereiche der linearen Algebra

Gauß-Algorithmus

Elementare Zeilenoperationen

1. Zeilen vertauschen: $Z_i \leftrightarrow Z_j$
2. Zeile mit Skalar multiplizieren: $Z_i \leftarrow \lambda Z_i$ $(\lambda \neq 0)$
3. Vielfaches einer Zeile zu anderer addieren: $Z_i \leftarrow Z_i + \lambda Z_j$

Ziel: Zeilenstufenform

$$\begin{pmatrix} \boxed{*} & * & * & * \\ 0 & \boxed{*} & * & * \\ 0 & 0 & \boxed{*} & * \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Boxed = Pivotelement (erste Nicht-Null in Zeile)

Beispiel: Schritt für Schritt

$$\left(\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{array}\right)$$

$Z_2 \leftarrow Z_2 + \frac{3}{2} Z_1$:

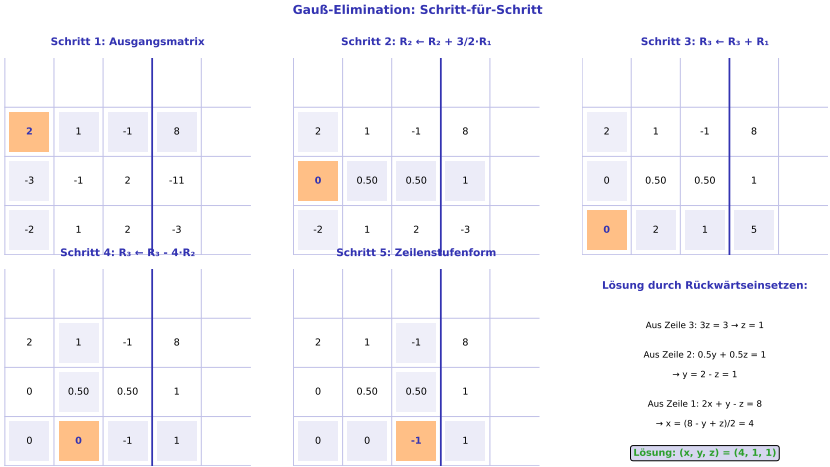
$$\left(\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ -2 & 1 & 2 & -3 \end{array}\right)$$

$Z_3 \leftarrow Z_3 + Z_1$:

$$\left(\begin{array}{ccc|c} 2 & 1 & -1 & 8 \\ 0 & \frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 2 & 1 & 5 \end{array}\right)$$

Der Gauß-Algorithmus transformiert systematisch in Zeilenstufenform

Gauß-Elimination: Visualisierung



Schritt-für-Schritt Transformation zur Zeilenstufenform

Rangbestimmung

Definition Rang

- Rang(A) = Anzahl der Pivotelemente
- = Anzahl linear unabhängiger Zeilen
- = Anzahl linear unabhängiger Spalten
- = Dimension des Spaltenraums

Eigenschaften

- $\text{Rang}(A) \leq \min(m, n)$
- $\text{Rang}(A) = \text{Rang}(A^T)$
- $\text{Rang}(AB) \leq \min(\text{Rang}(A), \text{Rang}(B))$

Beispiel: Rangbestimmung

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

Nach Gauß-Elimination:

$$\begin{pmatrix} \boxed{1} & 2 & 3 \\ 0 & 0 & 0 \\ 0 & \boxed{-1} & -1 \end{pmatrix}$$

$\Rightarrow$ Rang(A) = 2

Interpretation:

- Nur 2 unabhängige Gleichungen
- Lösungsraum hat Dimension $n - \text{Rang} = 1$

Der Rang bestimmt die Dimension des Lösungsraums

Teil II: Kern und Bild

Kern einer linearen Abbildung

Definition Der Kern (Nullraum) einer linearen Abbildung $f : V \to W$ ist:

$$\text{Kern}(f) = \{\vec{v} \in V : f(\vec{v}) = \vec{0}\}$$

Für Matrix $A \in \mathbb{R}^{m \times n}$:

$$\text{Kern}(A) = \{\vec{x} \in \mathbb{R}^n : A\vec{x} = \vec{0}\}$$

Eigenschaften

- Kern ist ein Unterraum von V
- $\dim(\text{Kern}(A)) = n - \text{Rang}(A)$
- f injektiv $\Leftrightarrow \text{Kern}(f) = \{\vec{0}\}$

Berechnung des Kerns Löse homogenes System $A\vec{x} = \vec{0}$:

Beispiel: $A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \end{pmatrix}$

Nach Gauß:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Freie Variablen: x_2, x_3

Basis des Kerns:

$$\text{Kern}(A) = \text{span}\left\{ \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \right\}$$

$\dim(\text{Kern}(A)) = 2$

Der Kern beschreibt alle Vektoren, die auf Null abgebildet werden

Bild einer linearen Abbildung

Definition Das Bild einer linearen Abbildung $f : V \to W$ ist:

$$\text{Bild}(f) = \{f(\vec{v}) : \vec{v} \in V\} \subseteq W$$

Für Matrix $A \in \mathbb{R}^{m \times n}$:

$$\text{Bild}(A) = \{A\vec{x} : \vec{x} \in \mathbb{R}^n\}$$

= Spaltenraum von A

Eigenschaften

- Bild ist ein Unterraum von W
- $\dim(\text{Bild}(A)) = \text{Rang}(A)$
- f surjektiv $\Leftrightarrow \text{Bild}(f) = W$

Basis des Bildes Pivotspalten bilden eine Basis!

Beispiel: $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 6 & 8 \end{pmatrix}$

Nach Gauß:

$$\begin{pmatrix} \boxed{1} & 2 & 3 \\ 0 & 0 & \boxed{-1} \\ 0 & 0 & 0 \end{pmatrix}$$

Pivotspalten: 1 und 3
Basis des Bildes:

$$\text{Bild}(A) = \text{span}\left\{ \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 5 \\ 8 \end{pmatrix} \right\}$$

Das Bild beschreibt alle erreichbaren Vektoren

Dimensionssatz

Fundamentaler Satz Für lineare Abbildung $f : V \to W$:

$$\boxed{\dim(V) = \dim(\text{Kern}(f)) + \dim(\text{Bild}(f))}$$

Für Matrix $A \in \mathbb{R}^{m \times n}$:

$$n = \dim(\text{Kern}(A)) + \text{Rang}(A)$$

Interpretation

- Eingabedimension = Freiheitsgrade + Einschränkungen
- Je größer der Kern, desto kleiner das Bild
- Maximales Bild $\Rightarrow$ trivialer Kern

Beispiel

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 4 & -2 \\ -1 & -2 & 1 \end{pmatrix}$$

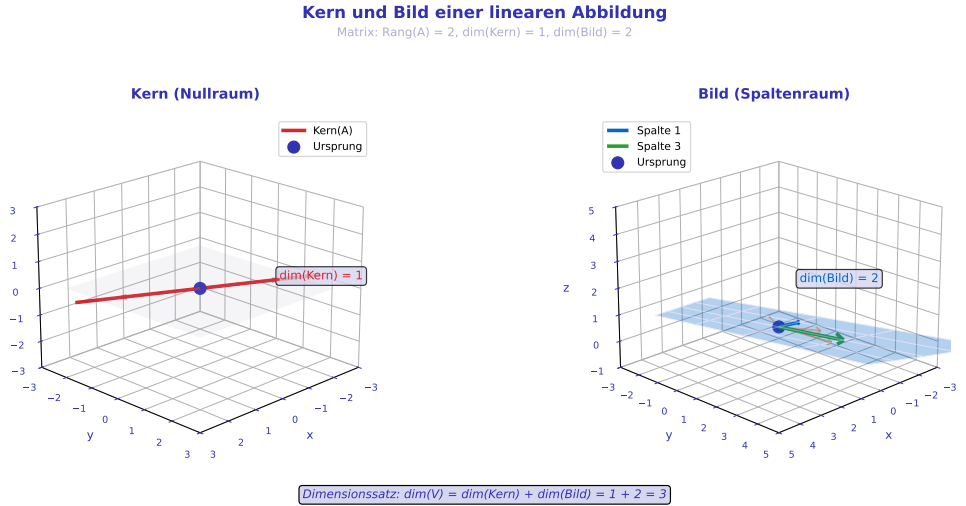
- $n = 3$ (Spalten)
- $\text{Rang}(A) = 1$ (nur eine unabhängige Zeile)
- $\dim(\text{Kern}(A)) = 3 - 1 = 2$
- $\dim(\text{Bild}(A)) = 1$

Verifikation: $2 + 1 = 3$ ✓

Geometrisch: Abbildung auf eine Linie durch den Ursprung

Der Dimensionssatz verbindet Kern und Bild fundamental

Kern und Bild: Visualisierung



Geometrische Darstellung von Kern (Nullraum) und Bild (Spaltenraum)

Teil III: Basiswechsel und Koordinatentransformation

Darstellungsmatrizen bei Basiswechsel

Setup

- Vektorraum V mit Basen B und B'
- Lineare Abbildung $f : V \to W$
- Basen C und C' von W

Darstellungsmatrizen

- $[f]_B^C$: Matrix von f bzgl. Basen B und C
- $[f]_{B'}^{C'}$: Matrix von f bzgl. Basen B' und C'

Transformationsformel

$$[f]_{B'}^{C'} = [id]_C^{C'} \cdot [f]_B^C \cdot [id]_{B'}^B$$

Basiswechselmatrizen

- $[id]_{B'}^B$: Basiswechsel von B' nach B
- Spalten = Koordinaten der B'-Vektoren in Basis B

Beispiel in $\mathbb{R}^2$

$$B = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

$$B' = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}$$

Basiswechselmatrix:

$$[id]_{B'}^B = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Darstellungsmatrizen ändern sich systematisch bei Basiswechsel

Koordinatentransformation

Koordinatenumrechnung Vektor $\vec{v}$ hat Koordinaten:

- $[\vec{v}]_B$ in Basis B
- $[\vec{v}]_{B'}$ in Basis B'

Umrechnungsformel

$$[\vec{v}]_B = [id]_{B'}^{B} \cdot [\vec{v}]_{B'}$$

$$[\vec{v}]_{B'} = ([id]_{B'}^{B})^{-1} \cdot [\vec{v}]_B$$

Merkhilfe

- Oberer Index = Zielraum
- Unterer Index = Ausgangsraum
- Indizes müssen "passen"

Konkretes Beispiel Vektor $\vec{v} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ in Standardbasis

Neue Basis: $B' = \left\{ \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$

Basiswechselmatrix:

$$P = [id]_{B'}^{B} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

Koordinaten in B':

$$[\vec{v}]_{B'} = P^{-1} \cdot \begin{pmatrix} 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

Verifikation:

$$2 \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} - 1 \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

✓

Koordinatentransformation ist essentiell für praktische Anwendungen

Teil IV: Grundlagen der Codierungstheorie

Motivation: Fehlerkorrektur

Problem

- Digitale Kommunikation ist fehleranfällig
- Rauschen, Interferenz, Datenverlust
- Kritisch bei: Satelliten, CDs, QR-Codes, RAM

Lösung: Redundanz

- Füge kontrollierte Redundanz hinzu
- Ermöglicht Fehlererkennung
- Ermöglicht Fehlerkorrektur

Trade-off

- Mehr Redundanz = bessere Korrektur
- Mehr Redundanz = weniger Effizienz

Beispiel: Paritätsbit

- Nachricht: 1011
- Mit Parität: 10111 (gerade Anzahl Einsen)
- Empfangen: 10101
- Fehler erkannt! (ungerade Anzahl)

Anwendungen

- ISBN/EAN Codes
- RAID Speichersysteme
- Mobilfunk (5G)
- Satellitenkommunkation
- QR-Codes
- DNA-Sequenzierung

Codierungstheorie macht moderne digitale Kommunikation erst möglich

Grundbegriffe der Codierungstheorie

Alphabet und Codes

- Alphabet: $\mathbb{F}_q$ (endlicher Körper)
- Häufig: $\mathbb{F}_2 = \{0, 1\}$ (binär)
- Codewort: Vektor $\vec{c} \in \mathbb{F}_q^n$
- Code: Menge von Codewörtern $C \subseteq \mathbb{F}_q^n$

Parameter eines Codes

- n: Blocklänge (Gesamtlänge)
- k: Dimension (Informationsbits)
- d: Minimaldistanz
- Notation: $[n, k, d]$-Code
- Rate: $R = k/n$

Hamming-Distanz Anzahl unterschiedlicher Positionen:

$$d_H(\vec{x}, \vec{y}) = |\{i : x_i \neq y_i\}|$$

Beispiel:

$$d_H(1011, 1101) = 2$$
$$d_H(0000, 1111) = 4$$

Minimaldistanz

$$d = \min_{\vec{c}_1, \vec{c}_2 \in C, \vec{c}_1 \neq \vec{c}_2} d_H(\vec{c}_1, \vec{c}_2)$$

Fehlerkorrektur-Fähigkeit

- Erkennt bis zu $d - 1$ Fehler
- Korrigiert bis zu $\lfloor \frac{d-1}{2} \rfloor$ Fehler

Die Hamming-Distanz ist die zentrale Metrik der Codierungstheorie

Lineare Codes

Definition Ein linearer Code C ist ein Unterraum von $\mathbb{F}_q^n$:

- $\vec{0} \in C$ (Nullwort ist Codewort)
- $\vec{c_1}, \vec{c_2} \in C \Rightarrow \vec{c_1} + \vec{c_2} \in C$
- $\vec{c} \in C, \alpha \in \mathbb{F}_q \Rightarrow \alpha\vec{c} \in C$

Vorteile linearer Codes

- Einfache Beschreibung durch Matrizen
- Effiziente Codierung/Decodierung
- Strukturierte Fehlerkorrektur
- Mathematisch elegant

Generatormatrix G

- $G \in \mathbb{F}_q^{k \times n}$
- Zeilen = Basis des Codes
- Codierung: $\vec{c} = \vec{m} \cdot G$
- $\vec{m} \in \mathbb{F}_q^k$ = Nachricht

Beispiel: $[4, 2, 2]$**-Code**

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

Codewörter:

$$(0,0) \cdot G = (0,0,0,0)$$
$$(1,0) \cdot G = (1,0,1,1)$$
$$(0,1) \cdot G = (0,1,0,1)$$
$$(1,1) \cdot G = (1,1,1,0)$$

Lineare Codes nutzen die Struktur von Vektorräumen

Teil V: Hamming-Codes

Hamming-Codes: Überblick

Richard Hamming (1950)

- Erster praktischer Fehlerkorrekturcode
- Perfekte 1-Fehler-korrigierende Codes
- Optimal für Single-Error-Correction

Hamming-Code Parameter

- Länge: $n = 2^r - 1$
- Dimension: $k = 2^r - r - 1$
- Minimaldistanz: $d = 3$
- Notation: $[2^r - 1, 2^r - r - 1, 3]$

Beispiele

- $r = 3$: $[7, 4, 3]$-Hamming-Code
- $r = 4$: $[15, 11, 3]$-Hamming-Code

Eigenschaften

- Korrigiert genau 1 Fehler
- Erkennt bis zu 2 Fehler
- Perfekt: nutzt Hamming-Schranke optimal
- Systematisch: Daten und Prüfbits getrennt

Hamming-Schranke Für t-fehlerkorrigierenden Code:

$$|C| \leq \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}$$

Hamming-Codes erreichen Gleichheit für $t = 1$!

Anwendung

- RAM mit ECC
- RAID-2 Speichersysteme

Hamming-Codes sind die einfachsten nicht-trivialen perfekten Codes

Generator- und Prüfmatrix

Systematische Generatormatrix

$$G = [I_k | P]$$

- I_k: $k \times k$ Einheitsmatrix
- P: $k \times (n-k)$ Paritätsmatrix
- Erste k Bits = Originalnachricht

Beispiel: $[7, 4, 3]$**-Hamming**

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Prüfmatrix (Parity Check)

$$H = [-P^T | I_{n-k}]$$

Eigenschaft: $G \cdot H^T = 0$

Beispiel: $[7, 4, 3]$**-Hamming**

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

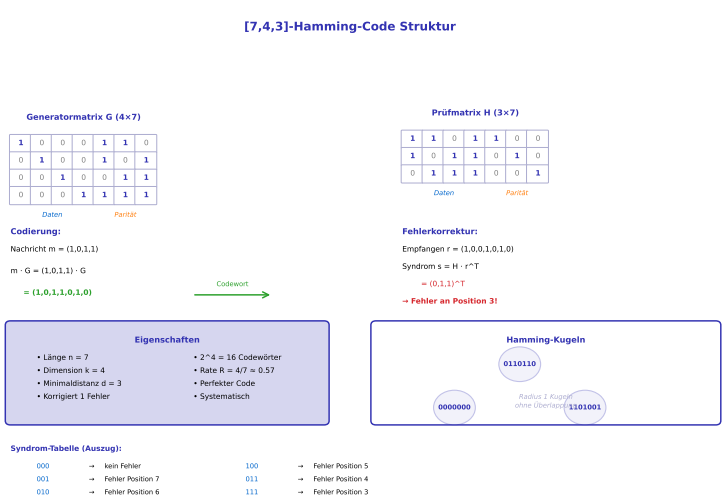
Syndrom Für empfangenes Wort $\vec{r}$:

$$\vec{s} = H \cdot \vec{r}^T$$

- $\vec{s} = \vec{0}$: kein Fehler
- $\vec{s} \neq \vec{0}$: zeigt Fehlerposition

Generator- und Prüfmatrix sind dual zueinander

Hamming-Code: Struktur-Visualisierung



Vollständige Struktur des [7,4,3]-Hamming-Codes mit Matrizen und Eigenschaften

Fehlerkorrektur mit Hamming-Codes

Syndrom-Decodierung

1. Empfange Wort $\vec{r}$
2. Berechne Syndrom: $\vec{s} = H \cdot \vec{r}^T$
3. Falls $\vec{s} = \vec{0}$: kein Fehler
4. Falls $\vec{s} \neq \vec{0}$:
   - $\vec{s}$ = Spalte i von H
   - Fehler an Position i
   - Korrigiere: $r_i \leftarrow r_i + 1 \pmod 2$

Warum funktioniert's?

- Alle Spalten von H sind verschieden
- Jedes Syndrom zeigt eindeutige Position
- Bei 1 Fehler: Syndrom = Fehlerspalte

Beispiel: $[7, 4, 3]$-**Code** Sende: $\vec{c} = (1, 0, 1, 0, 0, 1, 1)$
Fehler an Position 3: $\vec{r} = (1, 0, 0, 0, 0, 1, 1)$
Syndrom:

$$\vec{s} = H \cdot \vec{r}^T = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Dies ist Spalte 3 von H!
Korrektur: Bit 3 flippen

$$\vec{c} = (1, 0, 1, 0, 0, 1, 1)$$

✓

Implementierung

- Syndrom-Tabelle vorberechnen
- Lookup für schnelle Decodierung

Syndrom-Decodierung macht Hamming-Codes effizient implementierbar

Erweiterte Hamming-Codes

Zusätzliches Paritätsbit

- Füge Gesamt-Paritätsbit hinzu
- $[2^r - 1, 2^r - r - 1, 3] \to [2^r, 2^r - r - 1, 4]$
- Beispiel: $[7, 4, 3] \to [8, 4, 4]$

Vorteile

- Minimaldistanz 4 statt 3
- Erkennt 3 Fehler (statt 2)
- Korrigiert 1, erkennt zusätzlich 2
- SEC-DED: Single Error Correction, Double Error Detection

Anwendung

- Computer-RAM (ECC-Speicher)
- Kritische Systeme

Konstruktion Erweiterte Prüfmatrix:

$$H_{ext} = \begin{pmatrix} H \\ 1 \cdots 1 \end{pmatrix}$$

Decodierung

- 1 Fehler: ungerades Syndrom-Gewicht
- 2 Fehler: gerades Syndrom-Gewicht $\neq 0$
- 0 Fehler: Syndrom $= 0$

Beispiel: SECDED-Code

$$H_{[8,4,4]} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Erweiterte Hamming-Codes sind Standard in modernen Computern

Woche 3 Zusammenfassung

Gauß-Algorithmus

- Systematische Lösung von LGS
- Zeilenstufenform durch Elementaroperationen
- Rangbestimmung

Kern und Bild

- Kern = Nullraum der Abbildung
- Bild = erreichbare Vektoren
- Dimensionssatz: $\dim(V) = \dim(\text{Kern}) + \dim(\text{Bild})$

Basiswechsel

- Darstellungsmatrizen transformieren
- Koordinatenumrechnung
- Systematische Behandlung

Codierungstheorie

- Fehlerkorrektur durch Redundanz
- Hamming-Distanz als Metrik
- Lineare Codes als Unterräume

Hamming-Codes

- Perfekte 1-Fehler-Korrektur
- Generator- und Prüfmatrix
- Syndrom-Decodierung
- Praktische Anwendung in Computern

Kernkompetenzen

- Theorie trifft Praxis
- Von abstrakten Räumen zu konkreten Codes

Diese Woche verbindet abstrakte Algebra mit realen Anwendungen

Anhänge

Anhang A: Gauß-Algorithmus - Übungen

Aufgabe 1 Lösen Sie:

$$\begin{cases} 2x + 3y - z = 1 \\ x - y + 2z = 3 \\ 3x + 2y + z = 4 \end{cases}$$

Aufgabe 2 Bestimmen Sie den Rang:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 7 & 9 \\ 3 & 6 & 10 & 13 \\ 1 & 2 & 4 & 5 \end{pmatrix}$$

Aufgabe 3 Für welche $a \in \mathbb{R}$ hat das System

$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & a \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ b \end{pmatrix}$$

- keine Lösung?
- genau eine Lösung?
- unendlich viele Lösungen?

Aufgabe 4 Zeigen Sie: Elementaroperationen ändern den Rang nicht.

Übung macht den Meister beim Gauß-Algorithmus

Anhang B: Kern und Bild - Berechnungen

Beispiel 1: Kern bestimmen

$$A = \begin{pmatrix} 1 & 2 & -1 & 0 \\ 2 & 4 & -1 & 1 \\ 3 & 6 & -2 & 1 \end{pmatrix}$$

Lösung:

1. Gauß-Elimination
2. Freie Variablen identifizieren
3. Basis aufstellen

Beispiel 2: Bild bestimmen

$$B = \begin{pmatrix} 1 & 3 \\ 2 & 6 \\ -1 & -3 \end{pmatrix}$$

Spalten sind linear abhängig! $\dim(\text{Bild}(B)) = 1$

Beispiel 3: Dimensionssatz Gegeben: $f : \mathbb{R}^5 \to \mathbb{R}^3$ mit Rang 2
Bestimme:

- $\dim(\text{Kern}(f)) = ?$
- Ist f injektiv?
- Ist f surjektiv?

Beispiel 4: Projektionsmatrix

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- Kern = ?
- Bild = ?
- $P^2 = ?$

Kern und Bild charakterisieren lineare Abbildungen vollständig

Anhang C: Hamming-Code Tabellen

$[7, 4, 3]$-Hamming-Code

Alle 16 Codewörter:

Daten	Codewort
0000	0000000
0001	0001111
0010	0010011
0011	0011100
0100	0100101
0101	0101010
0110	0110110
0111	0111001
1000	1000110
1001	1001001
1010	1010101
1011	1011010
1100	1100011
1101	1101100
1110	1110000
1111	1111111

Syndrom-Tabelle

Syndrom	Fehlerposition
000	kein Fehler
001	Position 7
010	Position 6
011	Position 4
100	Position 5
101	Position 1
110	Position 2
111	Position 3

Eigenschaften

- Jedes Codewort hat Gewicht ≥ 3
- Je zwei Codewörter unterscheiden sich in ≥ 3 Positionen
- Perfekte Kugelpackung im $\mathbb{F}_2^7$

Diese Tabellen zeigen die vollständige Struktur des $[7, 4, 3]$-Codes