28.06.2021

BCH - Codes

- von Dominic Witt -

- Proseminar: Kodierungstheorie - Prof. Dr. B. Klopsch -
- Lit.: A First Course in Coding Theory, Raymond Hill -

- Bose, Ray-Chaudhuri (1960); Hocquenghem (1959)
- Hardy: Zahlentheorie hat keine nützliche Anwendung! (1940); Gegenbeispiele von Levinson (1970); ausgerechnet Ramanujan lieferte Arbeiten, die heute Anwendung in Kodierungstheorie finden (1912)

Theorem 11.1 K Körper, paarw. versch. $a_1, \dots, a_r \in K \setminus \{0\}$.

Die sog. Vandermonde-Matrix (auch Polynominterpolation)

$$A = \begin{pmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_r \\ a_1^2 & a_2^2 & \dots & a_r^2 \\ \vdots & \vdots & & \vdots \\ a_1^{r-1} & a_2^{r-1} & \dots & a_r^{r-1} \end{pmatrix}$$

ist invertierbar, d.h. $\det A \neq 0$. (LA)

Genauer gilt: $\det A = \prod_{i>j} (a_i - a_j)$

Beweis: Induktion über r. Beachte: $A \in K^{r \times r}$.

- $r = 1 \Rightarrow A = (1) = 1$ ✓
- Die Behauptung gelte für $r-1$.

• $\det A = \det \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & a_2 - a_1 & \cdots & a_r - a_1 \\ 0 & a_2(a_2 - a_1) & \cdots & a_r(a_r - a_1) \\ 0 & a_2^2(a_2 - a_1) & \cdots & a_r^2(a_r - a_1) \\ \vdots & \vdots & & \vdots \\ 0 & a_2^{r-2}(a_2 - a_1) & \cdots & a_r^{r-2}(a_r - a_1) \end{pmatrix}$

$(i+1)$te Zeile $- a_1 \cdot$ $i$te Zeile

$$= (a_2 - a_1)(a_3 - a_1) \cdots (a_r - a_1) \cdot \det \begin{pmatrix} 1 & 1 & \cdots & 1 \\ a_2 & a_3 & \cdots & a_r \\ a_2^2 & a_3^2 & \cdots & a_r^2 \\ \vdots & \vdots & & \vdots \\ a_2^{r-2} & a_3^{r-2} & \cdots & a_r^{r-2} \end{pmatrix}$$

Induktionsvor.

$$= (a_2 - a_1)(a_3 - a_1) \cdots (a_r - a_1) \cdot \prod_{r \geq i > j \geq 2} (a_i - a_j)$$

$$= \prod_{r \geq i > j \geq 1} (a_i - a_j) .$$

• Da $a_1, \dots, a_r$ paarw. versch. sind, gilt also $\det A \neq 0$. //

<u>Theorem 11.2</u> Sei $A \in K^{r \times r}$ invertierbar.

Die r Spalten von A, aufgefasst als Elemente der K-VR K^r, sind linear unabhängig (über K).

<u>Beweis</u>: lineare Algebra

Wiederholung:

$q = p^f$ Primzahlpotenz

$V(n, q)$ n-dimensionaler VR über $GF(q)$

linearer Code C = Untervektorraum von $V(n, q)$
↓ k-dimensional

Codewort $x_1 x_2 x_3 \dots x_n \in [n, k]$-Code

Dual-Code $C^\perp = \{ v \in V(n, q) \mid v \cdot u = 0 \ \forall \ u \in C \}$

Paritätsprüfungsmatrix H = Erzeugermatrix von $C^\perp$
↳ Zeilen bilden Basis für $C^\perp$

$\rightarrow C = \{ x \in V(n, q) \mid x H^T = 0 \}$

$\in V(n, q)$

(Hamming-) Abstand $d(x, y)$ = # unterssch. Stellen

$\rightarrow$ Abstandsfunktion

Minimalabstand $d(C) = \min\limits_{x \neq y \in C} d(x, y) = d$

$\rightarrow [n, k, d]$-Code

Beispiel 11.3 - Ein 2-Fehler-korrigierender Code

Sei C der lineare $[10,6]$-Code über $GF(11)$ definiert durch die Paritätsprüfungsmatrix

$$H = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & 10 \\ 1 & 2^2 & 3^2 & \cdots & 10^2 \\ 1 & 2^3 & 3^3 & \cdots & 10^3 \end{pmatrix}.$$

Bemerkungen: Mache C zu einem Dezimalcode D durch

(a) Auslassen aller Codewörter, die das Symbol 10 enthalten:

$$D = \left\{ x_1 x_2 \cdots x_{10} \in (F_{10})^{10} \;\middle|\; \sum_{i=1}^{10} x_i \equiv_{11} \sum_{i=1}^{10} i x_i \equiv_{11} \sum_{i=1}^{10} i^2 x_i \equiv_{11} \sum_{i=1}^{10} i^3 x_i \equiv_{11} 0 \right\}$$

wobei $F_{10} = \{0, 1, 2, \ldots, 9\}$

(b) Nach Theorem 8.4 (C, lin. $[n,k]$-Code über $GF(q)$ mit Paritätsprüfungsmatrix H, hat Minimalabstand d $\Leftrightarrow$ beliebige $d-1$ Spalten von H sind linear unabhängig, aber es ex. eine Auswahl von d linear abhängigen Spalten von H.) und den Theoremen 11.1 und 11.2 hat C den Minimalabstand 5. Theorem 1.9 (C kann bis zu t Fehler korrigieren, wenn $d(C) \geq 2t+1$ gilt.) besagt nun: C ist ein 2-Fehler-korrigierender Code.

Wiederholung: Für $y \in V(n,q)$ heißt

$$S(y) = y H^T$$ Syndrom von y.

Also: $S(y) = 0 \iff y \in C$. Und weiter:

$$u + C = v + C \quad (\in V(n,q)/C)$$

$$\iff u - v \in C$$

$$\iff S(u-v) = 0 \iff S(u) = S(v).$$

S ist lineare Abb.

Beispiel 11.3, Fortsetzung

Sei $x = x_1 x_2 \cdots x_{10}$ das vom Sender übertragene Codewort $\in C$ und $y = y_1 y_2 \cdots y_{10}$ das beim Empfänger angekommene Codewort $\in V(n,q)$.

Wir nehmen an, x und y stimmen an Stelle i und j nicht überein (und sonst schon) und $-x+y$ habe an Stelle i den Eintrag (a), an Stelle j den Eintrag (b) und an allen anderen Stellen Eintrag 0.

$$\to S(y) = y H^T = \Big(\underbrace{\sum_{i=1}^{10} y_i}_{S_1}, \underbrace{\sum_{i=1}^{10} i\, y_i}_{S_2}, \underbrace{\sum_{i=1}^{10} i^2 y_i}_{S_3}, \underbrace{\sum_{i=1}^{10} i^3 y_i}_{S_4} \Big)$$

Weil $x \in C$ ist, gilt $S(y) = S(-x+y)$, also:

$$\left| \begin{array}{rcl} a + b & = & S_1 \\ ai + bj & = & S_2 \\ ai^2 + bj^2 & = & S_3 \\ ai^3 + bj^3 & = & S_4 \end{array} \right|$$

→ Müssen nicht-lineares Gleichungssystem lösen.

$$\rightarrow \left| \begin{array}{l} b(i-j) = iS_1 - S_2 \\ bj(i-j) = iS_2 - S_3 \\ bj^2(i-j) = iS_3 - S_4 \end{array} \right|$$

$$\rightarrow (iS_2 - S_3)^2 = (iS_1 - S_2)(iS_3 - S_4)$$

$$\Leftrightarrow \underbrace{(S_2^2 - S_1 S_3)}_{P} i^2 + \underbrace{(S_1 S_4 - S_2 S_3)}_{Q} i + \underbrace{S_3^2 - S_2 S_4}_{R} = 0$$

Aus Symmetriegründen auch: $Pj^2 + Qj + R = 0$.

Also: Stellen der Fehler bekommen als Lösungen von $PX^2 + QX + R = 0$. Schließlich a, b bestimmen mit $\left| \begin{array}{l} a + b = S_1 \\ ai + bj = S_2 \end{array} \right|$ (LGS).

Falls nur ein Fehler in y vorhanden ist, etwa an Stelle i:

$a = S_1$, $ai = S_2$ $\quad (ai^2 = S_3,\ ai^3 = S_4)$

→ a und i einfach zu bestimmen; $P = Q = R = 0$

Algorithmus zur Bestimmung von i, j, a, b:

- Input: y
- $S(y) = (S_1, S_2, S_3, S_4)$ berechnen
- Falls $S(y) \neq 0$, berechne P, Q, R

- Falls $S(y) = 0$ ist, ist $y \in C$ und wir gehen davon aus, dass y keine Fehler enthält. (...$a = b = 0$)
  ($x \neq y$ möglich, aber y korrektes Codewort ...)
- Falls $S(y) \neq 0$ ist und $P = Q = R = 0$, gehen wir von einem einzigen Fehler aus, an Position $S_2 : S_1$ und der Größe S_1.
- Falls $P \neq 0$ und $R \neq 0$ ist und $Q^2 - 4PR \neq 0$ ein Quadrat in $GF(11)$ ist, gehen wir von 2 Fehlern aus, bei $i, j = (-Q \pm \sqrt{Q^2 - 4PR}) : (2P)$,
  der Größen $b = (iS_1 - S_2) : (i - j)$ bzw. $a = S_1 - b$.
- Sonst nehmen wir an, dass mindestens 3 Fehler in y aufgetaucht sind. (...)

Bemerkung: Die Berechnungen erfolgen stets modulo 11.
Dazu sollte man Tabellen/Listen für die Inversen, die Quadrate bzw. Wurzeln anlegen ...

Wollen nun Beispiel 11.3 verallgemeinern.

→ t-Fehler-korrigierender Code der Länge n über $GF(q)$

→ Voraussetzung: $2t + 1 \leq n \leq q - 1$

Theorem 1.9, $2t+1 \leq d \leq n$ (C ist $[n,1]$-Code $\Rightarrow d = n$ (bei H...)

$q = p$, siehe H...

Die folgende Klasse von Codes ist enthalten in der viel größeren Klasse der BCH-Codes. Mit den folgenden Codes wollen wir exemplarisch die wesentlichen Ideen demonstrieren:

Sei C der lineare Code über $GF(q)$, welcher durch die Paritätsprüfungsmatrix

$$H = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & n \\ 1 & 2^2 & 3^2 & \cdots & n^2 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{d-2} & 3^{d-2} & \cdots & n^{d-2} \end{pmatrix} \in GF(q)^{(d-1)\times n}$$

definiert ist; also $C = \{x \in V(n,q) \mid x H^T = 0\} \leq V(n,q)$.

Dabei gelte $d \leq n \leq q-1 = p-1$.

Beliebige $d-1$ Spalten von H liefern eine Vandermonde-Matrix.

Theoreme 11.1, 11.2 und 8.4 $\rightarrow$ C hat Minimalabstand d. Theorem 1.9 $\rightarrow$ t-Fehler-korr. Code, wenn $2t+1 \leq d$.

Ab jetzt sei d ungerade und t so, dass $2t+1 = d$ ist.

Dann hat H $2t$ Zeilen.

Vom Sender: $x = x_1 x_2 \cdots x_n$

Beim Empfänger: $y = y_1 y_2 \cdots y_n$

Fehler seien an den Stellen $X_1, X_2, \ldots, X_t$

mit den Größen $m_1, m_2, \ldots, m_t$

Wieder berechnen:

$$(S_1, S_2, \ldots, S_{2t}) = y\,H^T \quad , \text{ also}$$

$$S_j = \sum_{i=1}^{n} y_i\, i^{j-1} = \sum_{i=1}^{t} m_i\, X_i^{j-1}$$

vgl. S. 5 (... mod C ...)

für $j = 1, 2, \ldots, 2t$.

→ Gleichungssystem

$$(*) \quad \left|\begin{array}{lllll} m_1 & + m_2 & + \ldots + m_t & = S_1 \\ m_1 X_1 & + m_2 X_2 & + \ldots + m_t X_t & = S_2 \\ m_1 X_1^2 & + m_2 X_2^2 & + \ldots + m_t X_t^2 & = S_3 \\ & & \vdots & \\ m_1 X_1^{2t-1} & + m_2 X_2^{2t-1} & + \ldots + m_t X_t^{2t-1} & = S_{2t} \end{array}\right.$$

→ Für $t \geq 3$ Gleichungen zu kompliziert, um $2t-1$ der $2t$ Unbekannten auf die Weise wie zuvor in Beispiel 11.3 ($t = 2$) zu eliminieren

Ramanujan veröffentlichte 1912 eine andere Methode:
(Dabei ging es ihm nicht um Codierungstheorie ...)

Betrachte den (formalen) Ausdruck

$$\Phi(\theta) = \frac{m_1}{1 - X_1\theta} + \frac{m_2}{1 - X_2\theta} + \ldots + \frac{m_t}{1 - X_t\theta} \qquad (**)$$

Für $|\theta|$ klein genug haben wir für alle j:

$$\frac{m_j}{1 - X_j\theta} = m_j\left(1 + X_j\theta + X_j^2\theta^2 + \ldots\right)$$

(geometrische Reihe)

Also:

$$\Phi(\theta) = \sum_{j=1}^{t} m_j + \sum_{j=1}^{t} m_j X_j\,\theta + \sum_{j=1}^{t} m_j X_j^2\,\theta^2 + \ldots$$

Mit (*) erhalten wir:

$$\Phi(\theta) = S_1 + S_2\theta + S_3\theta^2 + \ldots + S_{2t}\theta^{2t-1} + \ldots$$

(**) umformen:

$$\Phi(\theta) = \frac{A_1 + A_2\theta + A_3\theta^2 + \ldots + A_t\theta^{t-1}}{1 + B_1\theta + B_2\theta^2 + \ldots + B_t\theta^t} \qquad (\dagger)$$

$$\Rightarrow (S_1 + S_2\theta + S_3\theta^2 + \ldots)(1 + B_1\theta + B_2\theta^2 + \ldots + B_t\theta^t) = A_1 + A_2\theta + A_3\theta^2 + \ldots + A_t\theta^{t-1}$$

Ausmultiplizieren zeigt:

$$\left.\begin{aligned} A_1 &= S_1 \\ A_2 &= S_2 + S_1 B_1 \\ A_3 &= S_3 + S_2 B_1 + S_1 B_2 \\ &\vdots \\ A_t &= S_t + S_{t-1} B_1 + S_{t-2} B_2 + \ldots + S_1 B_{t-1} \end{aligned}\right\} \quad (1)$$

$$\left.\begin{aligned} 0 &= S_{t+1} + S_t B_1 + S_{t-1} B_2 + \ldots + S_1 B_t \\ 0 &= S_{t+2} + S_{t+1} B_1 + S_t B_2 + \ldots + S_2 B_t \\ &\vdots \\ 0 &= S_{2t} + S_{2t-1} B_1 + S_{2t-2} B_2 + \ldots + S_t B_t \end{aligned}\right\} \quad (2)$$

Die $S_1, \ldots, S_{2t}$ sind bekannt. (2) ist LGS $\rightarrow$ $B_1, \ldots, B_t$ bestimmen. (1) liefert dann $A_1, \ldots, A_t$. Anschließend (†) nutzen und die rationale Funktion Φ in Partialbrüche zerlegen:

$$\Phi(\theta) = \frac{p_1}{1 - q_1\theta} + \frac{p_2}{1 - q_2\theta} + \ldots + \frac{p_t}{1 - q_t\theta}$$

Ein Vergleich mit (**) zeigt: $p_j = m_j$ und $q_j = x_j$ für alle $1 \leq j \leq t$. Wir haben (*) gelöst!

– ENDE –