Kapitel 7
Lineare Codes

In diesem Kapitel werden die bisher erarbeiteten Konzepte auf die Datenübertragung über einen nicht perfekten Kanal angewandt. Wir stellen uns vor, dass nacheinander Bits $x_1, x_2, x_3, \ldots$ über einen Kanal gesendet (oder auf einem Datenträger gespeichert) werden. Hierbei sind Fehler möglich: Mit einer gewissen Wahrscheinlichkeit (etwa $p = 10^{-6}$) wird ein Bit fehlerhaft übertragen bzw. gespeichert. Um trotzdem die korrekten Daten rekonstruieren zu können, oder um zumindest mit großer Wahrscheinlichkeit auf einen Fehler aufmerksam zu werden, schickt man die Daten mit einer gewissen Redundanz.

Die naivste Idee ist hierbei das Wiederholen: Alle Daten werden zweimal gesendet (oder 3,4, ... mal). Bei Einteilung in Viererblocks wird also statt (x_1, x_2, x_3, x_4) das „Wort“ $(x_1, x_2, x_3, x_4, x_1, x_2, x_3, x_4)$ gesendet.

Als allgemeinen Rahmen wollen wir die folgende Situation betrachten: Ein Bit wird als ein Element des Körpers $K = \mathbb{F}_2 (= \mathbb{Z}/2\mathbb{Z})$ modelliert. Wir können jedoch auch Elemente eines anderen (endlichen) Körpers K betrachten. Der zu sendende Bit-Strom wird in Blocks der Länge k zerlegt, z.B. $k = 4$. Statt $(x_1, \ldots, x_k) \in K^k$ wird $(c_1, \ldots, c_n) \in K^n$ gesendet (bzw. gespeichert). Hierbei gibt es eine Zuordnung $(x_1, \ldots, x_k) \mapsto (c_1, \ldots, c_n)$. Diese ist häufig linear, d.h. gegeben durch eine Matrix $G \in K^{n \times k}$, also:

$$\begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = G \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix}.$$

(Man beachte, dass wir hier je nach Bequemlichkeit Zeilen- und Spaltenvektoren schreiben.) Der gesendete Vektor $(c_1, \ldots, c_n)$ heißt **Codewort**, und $(x_1, \ldots, x_k)$ heißt **Informationswort**. G heißt **Generatormatrix**. Die Menge

$$C := \left\{ G \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} \;\middle|\; \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} \in K^k \right\}$$

aller Codewörter bildet einen Unterraum des K^n. Eine solche Datenübertragung ist nur sinnvoll, wenn die Zuordnung des Codeworts zu einem Datenwort injektiv ist. Das inhomogene LGS $G \cdot x = c$ muss also für alle $c \in C$ eindeutig lösbar sein, also $\operatorname{rg}(G) = k$. Aus unserem Test auf lineare Unabhängigkeit auf Seite 32 folgt, dass die Spalten von G linear unabhängig sind. Diese Spalten erzeugen C, also folgt

$$\dim(C) = k.$$

Ausgehend von dieser Situation machen wir folgende Definition:

Definition 7.1. *Ein* **linearer Code** *ist ein Unterraum $C \subseteq K^n$. Mit $k := \dim(C)$ bezeichnen wir C auch als einen (n,k)-Code. Die* **Länge** *von C ist n. Die* **Informationsrate** *ist k/n, die* **Redundanz** *ist $n-k$.*

Bei der Definition fällt auf, dass die Abbildung $K^k \to K^n$ nicht in die Definition des Codes aufgenommen wird. Für die meisten Fragestellungen der Codierungstheorie ist diese nämlich unerheblich. Als Generatormatrix eines Codes C kann man jede Matrix nehmen, deren Spalten eine Basis von C bilden. Wir bemerken noch, dass bisweilen auch nicht-lineare Codes betrachtet werden.

Beispiel 7.2. (1) Die Generatormatrix

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

liefert den *Wiederholungscode*, bei dem alles einmal wiederholt wird. Dies ist ein (8,4)-Code, die Informationsrate ist also 1/2. Falls bei der Übertragung höchstens ein Fehler auftritt, wird dies beim Empfang festgestellt. Der Fehler kann jedoch nicht korrigiert werden. Man spricht von einem *1-fehlererkennenden Code.*

(2) Der sogenannte *Parity-Check-Code* ist gegeben durch die Generatormatrix

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

Als Abbildung kann man ihn als $(x_1, \ldots, x_4) \mapsto (x_1, \ldots, x_4, x_1+x_2+x_3+x_4)$ definieren. Dies ist ein (5,4)-Code. Falls einer oder 3 Fehler auftreten, wird dies erkannt. Also ist auch dieser Code *1-fehlererkennend*. Aber seine Informationsrate ist mit 4/5 höher als die des Wiederholungscodes. Der Parity-Check-Code ist wohl eine der ältesten Ideen der Informatik.

(3) Es ist auch möglich, jedes Informationswort dreimal zu senden. Der entsprechende Code hat die Generatormatrix

$$G = \begin{pmatrix} I_4 \\ I_4 \\ I_4 \end{pmatrix} \in K^{12\times 4}.$$

Dies ist ein (12,4)-Code. Falls höchstens ein Fehler auftritt, kann man diesen nach Empfang korrigieren. Man spricht von einem *1-fehlerkorrigierenden Code*. ◁

Das *Dekodieren* läuft folgendermaßen ab: Das empfangene Wort $c' = (c'_1, \ldots, c'_n)$ kann sich von dem gesendeten Wort c durch Übertragungsfehler unterscheiden. Falls c' ein Codewort ist, also $c' \in C$, so wird $c = c'$ angenommen, denn dann ist der wahrscheinlichste Fall, dass kein Fehler auftrat. In diesem Fall wird durch das Auflösen des LGS $G \cdot x = c'$ das (wahrscheinliche) Informationswort $x \in K^k$ ermittelt. Interessanter ist der Fall $c' \notin C$. Es wird (wieder) mit der Annahme gearbeitet, dass die Anzahl der Fehlerbits mit großer Wahrscheinlichkeit klein ist. Also sucht man ein Codewort $c'' \in C$, das sich von c' an möglichst wenig Koordinaten unterscheidet. Falls es genau ein solches c'' gibt, wird $c = c''$ angenommen und $x \in K^k$ mit $G \cdot x = c''$ ausgegeben. Andernfalls wird eine Fehlermeldung ausgegeben: dann ist sinnvolles Dekodieren nicht möglich. Die Güte eines Codes entscheidet sich darin, dass dieser Fall möglichst vermieden wird, und dass korrektes Dekodieren ($c'' = c$) mit möglichst hoher Wahrscheinlichkeit passiert.

Definition 7.3. *Für $c = (c_1, \ldots, c_n) \in K^n$ ist*

$$w(c) := \left| \left\{ i \in \{1, \ldots, n\} \,\middle|\, c_i \neq 0 \right\} \right|$$

das **Hamming-Gewicht** *von c. Für $c, c' \in K^n$ ist*

$$d(c, c') := w(c - c') = \left| \left\{ i \in \{1, \ldots, n\} \,\middle|\, c_i \neq c'_i \right\} \right|$$

der **Hamming-Abstand** *von c und c'. (Nebenbei: Dies ist eine* Metrik *auf K^n.) Für eine Teilmenge $C \subseteq K^n$ ist*

$$d(C) := \min \left\{ d(c, c') \,\middle|\, c, c' \in C,\ c \neq c' \right\}$$

der **Hamming-Abstand** *von C. (Falls $|C| \leq 1$, so setzen wir $d(C) := n+1$.) Falls C ein Unterraum ist, ergibt sich*

$$d(C) = \min\left\{ w(c) \,\middle|\, c \in C \setminus \{0\} \right\}.$$

Beispiel 7.4. (1) Der (8,4)-Wiederholungscode (Beispiel 7.2(1)) hat $d(C) = 2$.
(2) Der (5,4)-Parity-Check-Code (Beispiel 7.2(2)) hat ebenfalls $d(C) = 2$.
(3) Der (12,4)-Wiederholungscode (Beispiel 7.2(3)) hat $d(C) = 3$. ◁

Folgende Überlegung zeigt, dass der Hamming-Abstand entscheidend ist für die Güte eines Codes.

Es sei zunächst $d(C) = 2e + 1$ ungerade. Das (durch Übertragungsfehler bedingte) Ändern von höchstens e Bits in einem Codewort ergibt ein $c' \in K^n$ mit $d(c, c') \leq e$. Dann ist c das eindeutig bestimmte Codewort $c'' \in C$ mit $d(c'', c') \leq e$. Aus $d(c'', c') \leq e$ und $c'' \in C$ folgt nämlich $d(c'', c) \leq 2e$, also $c'' = c$ wegen der Annahme. Dies bedeutet, dass korrekt dekodiert wird, falls höchstens e Übertragungsfehler auftreten. Der Code ist also e-fehlerkorrigierend. (Bei mehr als e Fehlern ist allerdings eine misslungene oder gar falsche Dekodierung möglich.)

Nun sei $d(C) = 2e + 2$ gerade. Nach obigem Argument ist C auch e-fehlerkorrigierend. Zusätzlich gilt: Bei $e + 1$ Fehlern gibt es kein Codewort $c'' \in C$ mit $d(c'', c') \leq e$ (denn dann wäre $c'' \neq c$ und $d(c, c'') \leq d(c, c') + d(c', c'') \leq e + 1 + e < d(C)$, ein Widerspruch). Dies bedeutet, dass $e + 1$ Fehler erkannt werden können. Bei $e + 1$ Fehlern kann aber (in der Regel) nicht mehr dekodiert werden. Ein Code mit Hamming-Abstand $2e+2$ ist also e-fehlerkorrigierend und $(e + 1)$-fehlererkennend.

Wir fassen zusammen:

Satz 7.5. *Sei $C \subseteq K^n$ ein Code.*

(a) Falls $d(C) = 2e + 1$, so ist C e-fehlerkorrigierend.
(b) Falls $d(C) = 2e+2$, so ist C e-fehlerkorrigierend und $(e+1)$-fehlererkennend.

Alles, was wir über das Dekodieren und den Hamming-Abstand gesagt haben, gilt auch für nicht-lineare Codes. Nun erinnern wir uns, dass wir lineare Codes betrachten wollen. Außerdem beobachten wir, dass in allen bisherigen Beispielen die Generatormatrix die Form

$$G = \begin{pmatrix} I_k \\ A \end{pmatrix} \tag{7.1}$$

mit $A \in K^{(n-k)\times k}$ hat. Wir machen dies zu einer weiteren Voraussetzung und bilden die Matrix

$$P := \begin{pmatrix} -A & I_{n-k} \end{pmatrix} \in K^{(n-k)\times n}.$$

P hat den Rang $n - k$, und es gilt

$$P \cdot G = \begin{pmatrix} -A & I_{n-k} \end{pmatrix} \cdot \begin{pmatrix} I_k \\ A \end{pmatrix} = 0.$$

Hieraus folgt $P \cdot c = 0$ für alle $c \in C$. Andererseits hat die Lösungsmenge L des homogenen LGS $P \cdot x = 0$ nach Proposition 6.11 die Dimension $n-(n-k) = k = \dim(C)$. Wegen Korollar 6.16(b) folgt $L = C$. Wir halten fest, dass für $c \in K^n$ gilt:

$$c \in C \quad \Longleftrightarrow \quad P \cdot c = 0.$$

P heißt die **Parity-Check-Matrix**. Nebenbei sei erwähnt, dass für lineare Codes auch ohne die Voraussetzung (7.1) eine Parity-Check-Matrix existiert.

Beispiel 7.6. (1) Der (8,4)-Wiederholungscode (Beispiel 7.2(1)) hat die Parity-Check-Matrix

$$P = \begin{pmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} \in K^{4\times 8}.$$

(2) Der (5,4)-Parity-Check-Code (Beispiel 7.2(2)) hat die Parity-Check-Matrix

$$P = \begin{pmatrix} -1 & -1 & -1 & -1 & 1 \end{pmatrix} \in K^{1\times 5}.$$

Mit Hilfe der Parity-Check-Matrix kann man das Dekodierungsverfahren verbessern. Es sei $c' \in K^n$ das empfangene Wort. Den Unterschied von c und c' quantifizieren wir durch den (dem Empfänger nicht bekannten) *Fehlervektor* $f := c' - c \in K^n$. Es ergibt sich

$$P \cdot c' = P \cdot (c + f) = 0 + P \cdot f = P \cdot f.$$

Der Vektor $P \cdot c' \in K^{n-k}$ heißt das **Syndrom** von c'. Es misst, wie weit c' von einem Codewort abweicht. Nach obiger Gleichung haben empfangenes Wort und Fehlervektor das gleiche Syndrom. Das Dekodieren kann nun so geschehen: Man berechnet das Syndrom $P \cdot c'$. Nun sucht man ein $f \in K^n$, welches unter allen $f' \in K^n$ mit $P \cdot f' = P \cdot c'$ minimales Hamming-Gewicht hat. Falls $c' \in C$, so ergibt sich automatisch $f = 0$. Falls es ein eindeutig bestimmtes solches f gibt, setzt man $c'' := c' - f \in C$ und gibt $x \in K^k$ mit $G \cdot x = c''$ aus. Falls es kein eindeutiges f gibt, gibt man eine Fehlermeldung aus. Dies entspricht genau dem oben beschriebenen Dekodierungsverfahren. Da es nur $|K|^{n-k}$ mögliche Syndrome gibt, kann man das f (oder Fehlermeldung) zu jedem Syndrom in einer Tabelle speichern. Oft gibt es noch bessere Methoden zur Ermittlung von f. Dies ist in folgendem Beispiel der Fall.

Der (7,4)-Hamming-Code

Wir definieren nun den sogenannten (7,4)-Hamming-Code. Dieser zeigt, dass Codierungstheorie zu mehr in der Lage ist, als die bisherigen, relativ offensichtlichen Beispiele von Codes zu analysieren. Der Hamming-Code $C \subset \mathbb{F}_2^7$

wird durch die Generatormatrix

$$G = \begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 0&1&1&1 \\ 1&0&1&1 \\ 1&1&0&1 \end{pmatrix} \in \mathbb{F}_2^{7\times 4}$$

definiert, als Abbildung $\mathbb{F}_2^4 \to \mathbb{F}_2^7$ also $(x_1, \ldots, x_4) \mapsto (x_1, x_2, x_3, x_4, x_2 + x_3 + x_4, x_1 + x_3 + x_4, x_1 + x_2 + x_4)$. C ist ein (7,4)-Code, hat also höhere Informationsrate als der (8,4)-Wiederholungscode aus Beispiel 7.2(1). Die Parity-Check-Matrix ist

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

Welchen Hamming-Abstand hat C? Dazu müssen wir $w(c)$ für $c \in C \setminus \{0\}$ ermitteln. Die Bedingung $c \in C$ ist gleichbedeutend mit $P \cdot c = 0$. Gibt es ein solches c mit $w(c) = 1$? Dies würde bedeuten, dass (mindestens) eine der Spalten von P eine Nullspalte ist, was nicht der Fall ist. Gibt es ein $c \in \mathbb{F}_2^7$ mit $P \cdot c = 0$ und $w(c) = 2$? Dies würde bedeuten, dass es in P zwei Spalten gibt, die linear abhängig sind. Auch dies ist nicht der Fall! Es folgt also $d(C) > 2$. In diesem Argument zeigt sich die eigentliche Idee des Hamming-Codes: Man beginnt mit der Parity-Check-Matrix und stellt sie so auf, dass sie keine zwei linear abhängigen Spalten enthält. Hieraus folgt dann $d(C) > 2$. Die Generatormatrix G leitet man dann aus der Parity-Check-Matrix her. Da G selbst (sogar mehr als) einen Vektor von Gewicht 3 enthält, folgt

$$d(C) = 3.$$

Der (7,4)-Hamming Code ist also 1-fehlerkorrigierend. Damit hat er einerseits eine höhere Informationsrate, andererseits bessere Fehlerkorrektureigenschaften als der (8,4)-Wiederholungscode!

Das Dekodieren ist hier ganz besonders einfach: Es gibt nur acht mögliche Syndrome, nämlich alle Vektoren von $\mathbb{F}_2^3$. Wir können diese schreiben als $v_0 = 0, v_1, \ldots, v_7$, wobei v_i die i-te Spalte von P ist $(i > 0)$. Für v_0 ist der Nullvektor das Codewort kleinsten Gewichtes mit Syndrom v_0. Für v_i $(i > 0)$ ist dies der i-te Standardbasisvektor e_i, denn $P \cdot e_i = v_i$. Der vollständige Dekodieralgorithmus läuft also so ab: Man ermittelt das Syndrom $s := P \cdot c'$ des empfangenen Wortes $c' = (c'_1, \ldots, c'_7)$. Falls $s = v_i$ mit $1 \le i \le 4$, so gibt man $(x_1, \ldots, x_4) = (c'_1, \ldots, c'_4) + e_i$ aus (d.h. das i-te Bit wird geändert). Andernfalls gibt man $(x_1, \ldots, x_4) = (c'_1, \ldots, c'_4)$ aus. (Falls das Syndrom einer der Vektoren v_5, v_6, v_7 ist, so wird e_i mit $i > 4$ zu c' hinzuaddiert, aber dies

ändert $(x_1, \ldots, x_4)$ nicht.) In dem wahrscheinlichen Fall, dass bei der Übertragung höchstens ein Fehler auftritt, wird so das korrekte Informationswort ausgegeben.

Der Bauer-Code

Einen weiteren interessanten Code erhalten wir durch folgende Erweiterung des (7,4)-Hamming Codes: Wir hängen einfach zusätzlich noch ein Parity-Bit $c_8 = c_1 + \cdots + c_7$ an, d.h. wir benutzen die Abbildung

$$(x_1, \ldots, x_4) \mapsto (x_1, x_2, x_3, x_4, x_2{+}x_3{+}x_4, x_1{+}x_3{+}x_4, x_1{+}x_2{+}x_4, x_1{+}x_2{+}x_3).$$

Der hierdurch definierte Code C wird *Bauer-Code* (nach F. L. Bauer, Informatiker an der TU München) genannt. Es ist ein (8,4)-Code. Was ist der Hamming-Abstand $d(C)$? Auf jeden Fall mindestens 3, denn die ersten 7 Bits sind ja identisch mit dem Hamming-Code. Aber falls ein Wort $(c_1, \ldots, c_7)$ des Hamming-Codes das Gewicht 3 hat, so ist $c_1 + \cdots + c_7 = 1$, also hat das entsprechende Wort in C Gewicht 4. Wir erhalten $d(C) = 4$. Der Bauer-Code ist also 1-fehlerkorrigierend und 2-fehlererkennend. Er hat damit wesentlich bessere Eigenschaften als der (8,4)-Wiederholungscode.