Exkurs: Fehlerkorrigierende Codes

Lit.: MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes, North-Holland, Amsterdam 1978.
van Lint, J.H.: Introduction to Coding Theory (3. Aufl.), Springer 1999.

Problem: bei der Übermittlung von Daten entstehen Fehler. Gesucht ist eine Methode, die die (meisten) Fehler automatisch korrigiert.

Daten $\overset{\text{Kodierung}}{\longrightarrow}$ endl. Folge der Länge n von $0,1$ $\overset{\text{Übermittlung}}{\longrightarrow}$ fehlerhafter Folge von $0,1$ $\overset{\text{automat. Fehlerkorrektur}}{\longrightarrow}$ ursprüngl. Folge von $0,1$ $\overset{\text{Dekodierung}}{\longrightarrow}$ ursprüngl. Daten.

Das kann nur funktionieren, wenn bei der Kodierung nur spezielle Folgen von $0,1$ ("die Wörter des Codes") verwendet werden.

Mathematische Formulierung:

$$\{0,1\}^n = \{(x_1, \ldots, x_n) \mid x_1 \in \{0,1\}, \ldots, x_n \in \{0,1\}\}$$

Folgen von $0,1$ der Länge n, z.B. $(0,1,0,1,0,1,0) \in \{0,1\}^7$

Hamming-Abstand d von Elementen $x, y \in \{0,1\}^n$

$$d(x,y) = \#\{i \mid 1 \leq i \leq n \text{ und } x_i \neq y_i\} \quad (\Rightarrow d(x,y) \in \{0, \ldots, n\})$$

Dabei wird mit $\#M$ die Anzahl der Elemente einer Menge M bezeichnet.

Bsp.: $(n = 4)$ $x = (0,0,1,1)$, $y = (0,1,1,0) \Rightarrow d(x,y) = 2$.

Eigenschaften von d: $\forall x, y, z \in \{0,1\}^n$ gilt:

(i) $d(x,y) \geq 0$ und $d(x,y) = 0 \Leftrightarrow x = y$
(ii) $d(x,y) = d(y,x)$ ("Symmetrie")
(iii) $d(x,y) \leq d(x,z) + d(z,y)$ ("Dreiecksungleichung")

(i)-(iii) $\Leftrightarrow$ "d ist Abstandsfunktion"

Zu $x \in \{0,1\}^n$, $t \in \mathbb{N}$ sei

$$B(x,t) = \{y \in \{0,1\}^n \mid d(x,y) \leq t\}$$

der (d-)Ball um x vom Radius t.

Bem.: $B(x,1)$ enthält genau $(n+1)$ Elemente.

(E1) Def.:

(i) Ein Code der Länge n ist eine Teilmenge $C \neq \emptyset$ von $\{0,1\}^n$.
(ii) Ein Code C heißt t-fehlerkorrigierend $\Leftrightarrow$
Ist $x \in C, y \in C, x \neq y$, so gilt $B(x,t) \cap B(y,t) = \emptyset$.

Idee: Sei C t-fehlerkorrigierender Code. Als kodierte Nachrichten sind nur "Codewörter" $x \in C$ erlaubt.

$$x \in C \overset{\text{Übermittlung}}{\longrightarrow} \overline{x} \in \{0,1\}^n \overset{\text{automat. Korrektur}}{\longrightarrow} \overline{\overline{x}} \in C \text{ so, daß } d(\overline{x}, \overline{\overline{x}}) \text{ minimal.}$$

Dann: Ist $d(x, \overline{x}) \leq t$, so gilt $\overline{\overline{x}} = x$, da es nur ein $x \in C$ mit $d(\overline{x}, x) \leq t$ gibt.
D.h.: Sind bei der Übermittlung höchstens t Fehler aufgetreten, so ordnet die automatische Fehlerkorrektur der fehlerhaften Nachricht $\overline{x}$ das ursprüngliche Codewort x zu.

Bem.: 1) $C \subseteq \{0,1\}^n$ ist t-fehlerkorrigierend $\Leftrightarrow$ Ist $x \in C$, $y \in C$ und $x \neq y$, so gilt $d(x,y) \geq 2t+1$ (Dreiecksungleichung!)
2) $C \subseteq \{0,1\}^n$ 1-fehlerkorriegierend $\Rightarrow \#C \cdot (n+1) \leq 2^n (= \#\{0,1\}^n)$

Problem: Zeitaufwand Fehlerkorrektur $\approx \#C \cdot n \approx 2^n$

Lineare Codes

$\{0,1\}^n$ ist Vektorraum über dem Körper $\{0,1\} = \mathbb{Z}_2$

(E2) Def. $C \subseteq \{0,1\}^n$ heißt linearer Code $\Leftrightarrow$ C ist Unterraum von $\{0,1\}^n$
($\Leftrightarrow \underline{0} \in C$ und $\forall x, y \in C : x + y \in C$)

Sei $p \in \mathbb{N}$ und $n := 2^p - 1$ (z.B. $p = 5, n = 31 \Rightarrow 2^n = 3^{31} = 2 \cdot (2^{10})^3 = 2 \cdot (1024)^3 \approx 2 \cdot 10^9$)

Seien $v_1, \ldots, v_n$ die von $\underline{0} \in \{0,1\}^p$ verschiedenen Elemente von $\{0,1\}^p$ und $F : \{0,1\}^n \to \{0,1\}^p$ definiert durch

$$F(x_1, \ldots, x_n) = x_1 v_1 + x_2 v_2 + \ldots + x_n v_n \in \{0,1\}^p$$

Dann gilt für alle $x, y \in \{0,1\}^n : F(x+y) = F(x) + F(y)$.

Daraus folgt: $C := \{x \in \{0,1\}^n \mid F(x) = \underline{0}\}$ ist Unterraum (=linearer Code).

C heißt der n'te Hammingcode.

(E3) Satz. *Der n'te Hammingcode ist ein* 1*-fehlerkorrigierender Code der Länge n*

$$\textit{mit } \#C = \frac{2^n}{n+1} \textit{(“C ist perfekt”)}$$

Bew.: Seien $x, y \in C$, $x \neq y$. Zu zeigen: $d(x,y) \geq 3$

$$x, y \in C \Rightarrow \quad (*) \quad (x_1 - y_1)v_1 + \ldots + (x_n - y_n)v_n = \underline{0}.$$

Wegen $x \neq y$ existiert ein $i \in \{1, \ldots, n\} : x_i \neq y_i$. Dann $(x_i - y_i)v_i = v_i \neq \underline{0}$.
Wegen $(*)$ existiert ein $j \neq i$ mit $x_j \neq y_j$, also $(x_j - y_j)v_j = v_j$. Wenn alle anderen Komponenten von x und y übereinstimmten, so wäre $(*)$ die Gleichung

$$v_i + v_j = \underline{0}$$

Da in $\mathbb{Z}_2$ $1+1=0$ gilt, gilt $v+v=\underline{0}$ für alle $v \in \{0,1\}^n$. D.h. aus $v_i+v_j=\underline{0}$, folgt $v_i=v_j$, im Widerspruch dazu, daß die v_i alle verschieden sind. Also gibt es mindestens eine weitere Komponente, in der x und y nicht übereinstimmen, d.h. $d(x,y) \geq 3$, wie behauptet.

Ist $x \in \{0,1\}^n$ und $F(x) \neq \underline{0}$, so gilt $F(x)=v_i$ für ein $i \in \{1,\ldots,n\}$.
Daraus folgt $F(x-e_i)=F(x)-F(e_i)=\underline{0}$, d.h. $x-e_i \in C$ und $d(x,x-e_i)=1$.
Also $\bigcup\limits_{x\in C} B(x,1)=\{0,1\}^n$ und damit $\#C\cdot(n+1)=2^n$.

Algorithmus zur Korrektur <u>eines</u> Fehlers:

Sei $\overline{x} \in \{0,1\}^n$ die empfangene Nachricht. Berechne $F(\overline{x}) \in \{0,1\}^p$.

$$\begin{aligned} F(\overline{x})=\underline{0} \quad &\Rightarrow \quad \overline{\overline{x}} := \overline{x} \\ F(\overline{x})=v_i \quad &\Rightarrow \quad \overline{\overline{x}} := \overline{x}+e_i \end{aligned}$$

Zeitaufwand $\approx n^2 p$ $(= 31^2 \cdot 5 \approx 5 \cdot 10^3) \ll 2^n \approx 2 \cdot 10^9)$