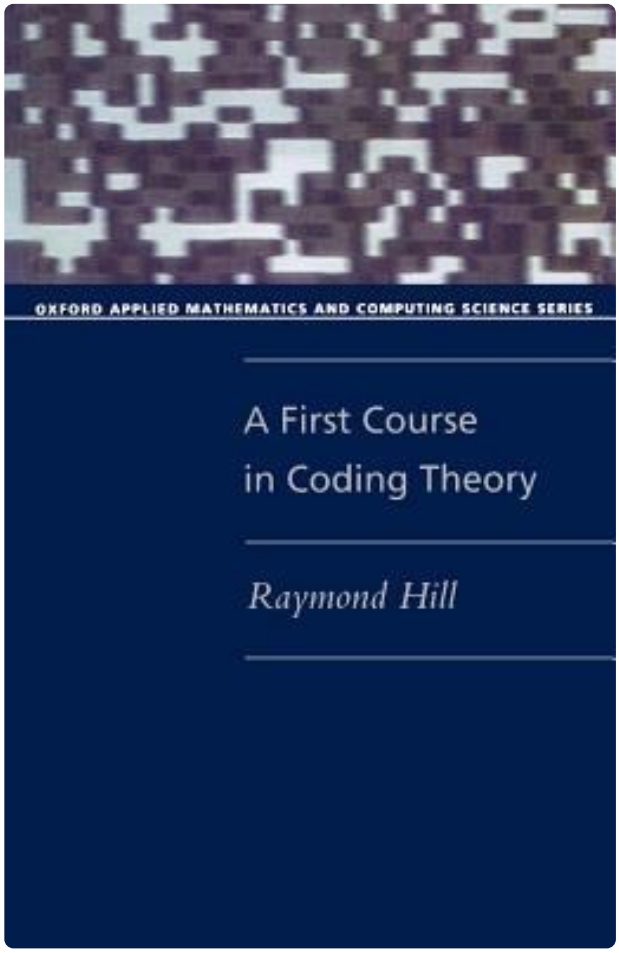
Proseminar
Kodierungstheorie 2021

bei Prof. Dr. Klopsch



http://www.goodreads.com/book/show/
746828.A_First_Course_in_Coding_Theory

Thema: Encoding and decoding with a linear code (Chapter 6)

Dea Dubovci, 17.05.21

1. Verschlüsseln eines linearen Codes

Def 1.1. Sei C ein $[n,k]-code$ über $GF(q)$ mit Generatormatrix G. Eine **Nachricht** ist ein k-Tupel von $V(k,q)$. Ein **Nachrichtsvektor** $u = u_1 \dots u_k$ wird verschlüsselt, indem man ihn lediglich von rechts mit G multipliziert.

Sei $G = [r_1 | r_2 | \dots | r_k]$, dann:

$$uG = \sum_{i=1}^{k} u_i r_i$$

d.h. uG ist ein **Codewort** von C, denn es ist eine Linearkombination der Spalten $[r_1 | r_2 | \dots | r_k]$ von G.

Bem. 1.2. Die **Verschlüsselungsfunktion** $u \to uG$ ordnet dem Vektorraum $V(k,q)$ einem k-dimensionalen Unterraum (nämlich dem Code C) zu.

Satz 1.3. Sei G in Standardform mit $G = [I_k | A]$, wobei $A = [a_{ij}]$ eine $k \times (n-k)$ Matrix ist. Dann wird der Nachrichtsvektor u durch

$$x = uG = x_1 x_2 \dots x_k x_{k+1} \dots x_n$$

verschlüsselt. Dabei sind

$x_i = u_i$, $1 \leq i \leq k$ die Nachrichten-ziffern und

$$x_{k+1} = \sum_{j=1}^{k} a_{ji} u_j, \quad 1 \leq i \leq n-k$$

sind Korrekturziffern. Letztere induzieren Redundanz, die der Nachricht angehangen wurden, um Störungen auszugleichen.

Bsp. 1.4. Sei C der binäre $[7,4]-code$ mit Generatormatrix in Standardform

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

Ein Nachrichtenvektor u wird kodiert durch:

$$uG = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_1 + u_2 + u_3 \\ u_2 + u_3 + u_4 \\ u_1 + u_2 + u_4 \end{pmatrix}$$

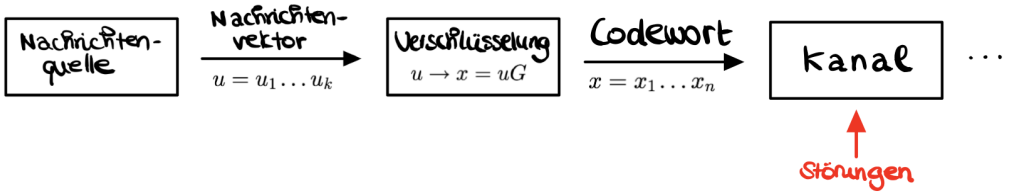
Beispielsweise gilt:

$$(0 \quad 0 \quad 0 \quad 0) \to (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$

$$(1 \quad 0 \quad 0 \quad 0) \to (1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1)$$

$$(1 \quad 1 \quad 0 \quad 0) \to (1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0)$$

Bem. 1.5. Allg. gilt für einen linearen Code für den Verschlüsselungsprozess folgendes Kommunikationsschema:



2. Entschlüsseln eines linearen Codes

Def. 2.1. Sei $x = x_1 \dots x_n$ ein Codewort, das durch den Kanal gesendet wird und als $y = y_1 \dots y_n$ ankommt. Wir definieren den Fehlervektor als $e := y - x = e_1 \dots e_n$.

Bem. 2.2. Der Entschlüssler hat nun zwei äquivalente Möglichkeiten:

i) Anhand von y erkennen, welches Codewort x übermittelt wurde

ii) Erkennen welcher Fehlervektor e entstanden ist.

Def 2.3. Sei C ein $[n,k] - code$ über $GF(q)$ und $a \in V(n,q)$. Dann heißt

$$a + C = \{a + x | \, x \in C\}$$

Nebenklasse von C.

Lemma 2.4. Sei $a + C$ eine Nebenklasse von C und $b \in a + C$. Dann:

$$b + C = a + C$$

Bew.: Da $b \in a + C$ gilt $b = a + x$ für gewisses $x \in C$. Sei nun $b + y \in b + C$. Dann:

$$\begin{aligned} b + y &= (a + x) + y \\ &= a + (x + y) \in a + C \end{aligned}$$

also $b + C \subseteq a + C$.

Sei nun $a + z \in a + C$. Dann:

$$\begin{aligned} a + z &= (b - x) + z \\ &= b + (z - x) \in b + C \end{aligned}$$

also $a + C \subseteq b + C$.

Insg. folgt $a + C = b + C$ □

Theorem 2.5. Sei C ein $[n, k] - code$ über $GF(q)$. Dann:

i) Jeder Vektor aus $V(q, n)$ ist in einer Nebenklasse von C enthalten.

ii) Jede Nebenklasse enthält genau q^k Vektoren.

iii) Zwei Nebenklassen sind entweder

disjunkt oder identisch.

Bew.: i) Falls $a \in V(n,q)$, dann:
$a = a + 0 \in a + C$

ii) Die Zuordnung von C nach $a+C$ def durch $x \to a + x \quad \forall x \in C$ ist offensichtlich injektiv, daher:
$|a + C| = |C| = q^k$

iii) Angenommen die Nebenklassen $a+C$ und $b+C$ überschneiden sich. Dann ex. ein Vektor v mit
$v \in (a + C) \cap (b + C)$.
Daher gilt für gewisse $x, y \in C$:
$v = a + x = b + y$.
Dann $b = a + (x - y) \in a + C$
und nach Lemma 2.4
$a + C = b + C$. □

Bsp. 2.6. Sei C der binäre $[4,2] - code$ mit Generatormatrix

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

z.B. $C = \{0000, 1011, 0101, 1110\}$.
Dann sind die Nebenklassen von C:

$$0000 + C = C$$
$$1000 + C = \{1000, 0011, 1101, 0110\}$$
$$0100 + C = \{0100, 1111, 0001, 1010\}$$
$$0010 + C = \{0010, 1001, 0111, 1100\}$$

Achtung:
Für $0001 + C = \{0001, 1010, 0100, 1111\}$
gilt $0001 + C = 0100 + C$. Das hätte man auch hervorsagen können, denn
$0001 \in 0100 + C$.

Def. 2.7. Der Vektor mit minimalem Gewicht (wdh. Gewicht $w(x) \cong$ Anzahl der Einträge in $x \neq 0$) einer Nebenklasse heißt Nebenklassenanführer.

Bem. 2.8. Falls mehrere Vektoren minimales Gewicht haben, so suchen wir uns einen bel. aus und bezeichnen ihn als Nebenklassenanführer.

Bem. 2.9. Wegen Theorem 2.5. gilt, dass $V(n,q)$ in disjunkte Nebenklassen von C zerlegt werden kann:

$$V(n,q) = (0 + C) \cup (a_1 + C) \cup \ldots \cup (a_s + C)$$

mit $s = q^{n-k} - 1$. Nach Lemma 2.4. können wir $0, a_1, \ldots, a_s$ als Nebenklassenanführer setzen.

Def. 2.10. Ein **Standardarray** für einen $[n,k] - code$ C ist ein $q^{n-k} \times q^k$-array mit allen Vektoren in $V(n,q)$, in denen die erste Reihe aus dem Code C mit 0 ganz außen links besteht und die restlichen Reihen sind die Nebenklassen $a_i + C$, wobei diese alle in korrespondierender Reihenfolge mit den Nebenklassenanführer links angeordnet sind.

Algo. 2.11 **Schritt 1:** Erstelle eine Liste aus den Codewörtern von C, beginnend mit 0 als erste Reihe.

Schritt 2: Wähle bel. Vektor $a_1 \notin (0 + C)$ mit minimalem Gewicht. Setze die Nebenklasse $a_1 + C$ als zweite Reihe, indem man 0 unter a_1 und

$a_1 + x$ unter $x \forall x \in C$ setzt.

Schritt 3: Aus den Vektoren, die nicht in den vorherigen Reihen enthalten sind, wählt man ein bel. a_2 mit minimalem Gewicht und ordnet die Nebenklasse $a_2 + C$ wie in Schritt 2 beschrieben an.

Schritt 4: Wiederhole dieses Vorgehen bis alle Nebenklassen sortiert sind und jeder Vektor aus $V(n, q)$ genau einmal auftaucht.

Bsp. 2.12. Ein Standardarray für den Code aus Bsp. 2.6. ist

Codewörter { 0000	1011	0101	1110
1000	0011	1101	0110
0100	1111	0001	1010
0010	1001	0111	1100

Nebenklassenanführer

Bem. 2.13. In einem Standardarray ist jeder Eintrag die Summe aus dem Codewort der Zeile darunter und dem Nebenklassenanführer dieser.

Bem. 2.14. Der Entschlüssler nutzt den Standardarray wie folgt: Wenn z.B. $y = 1111$ ankommt, wird dessen Position im array bestimmt. Dann entscheidet der Entschlüssler, dass der Fehlervektor e der Nebenklassenanführer 0100 ist, den man ganz links von y findet. Nun ist y entschlüsselt als das Codewort $x = y - e = 1011$. Kurzgesagt: Man entschlüsselt einen Vektor als das Codewort über der Spalte, die y enthält.

Bem. 2.15. Die Fehlervektoren, die korrigiert werden, sind gerade die Nebenklassenanführer, unabhängig davon, welches Codewort übermittelt wird. Indem man einen Vektor von minimalem Gewicht in jeder Nebenklasse als Nebenklassenanführer wählt, stellen wir sicher, dass der Standardarray, den wir zum entschlüsseln nutzen, ein „Nächster-

Nachbar-Schema" ist.

Bsp. 2.16. In Bsp. 2.12 wird ein einfacher Fehler nur korrigiert, wenn er in den ersten drei Stellen auftaucht, nicht aber in der vierten Stelle.

Nachricht	Codewort	Kanal + Störung	erh. Vektor	entschl. Wort	erh. Nachricht
01	0101	0101	0001	0101	01
01	0101	0101	0100	0000	00

Bem. 2.17. Im zweiten Fall sehen wir, dass Kanal und Störung das Codewort nicht beeinflusst haben, wir aber trotzdem die falsche Nachricht 00 erhalten haben. In diesem Fall hat das Anhängen der Redundanz mehr Schaden angerichtet, als nützlich zu sein.
Um aber eine sinnvolle Bewertung für einen „guten Code" zu bekommen, müssen wir die Wahrscheinlichkeit kennen mit der ein erhaltener Vektor mittels dem Standararray korrekt entschlüsselt wird.

3. Die Wahrscheinlichkeit einer Fehlerkorrektur

Bem. 3.1. Der Einfachheitshalber begrenzen wir uns in diesem Kapitel auf binäre lineare Codes. Wir nehmen an, dass der Kanal binär symmetrisch mit Zeichenfehlerwahrscheinlichkeit p verteilt ist. In einer vorherigen Sitzung haben wir festgehalten, dass die Wahrscheinlichkeit, dass der Fehlervektor ein gegebener Vektor mit Gewicht i ist, $p^i (1-p)^{n-i}$ beträgt ($\Rightarrow$ Binomialverteilung).

Theorem 3.2. Sei C ein binärer $[n,k]-code$. Für $i = 0, \ldots, n$ sei α_i die Anzahl der Nebenklassenanführer mit Gewicht i. Dann beträgt die Wahrscheinlichkeit, dass ein erhaltener Vektor, der mit einem Standardarray entschlüsselt wurde, korrekt ist

$$P_{corr}(c) := \sum_{i=0}^{n} \alpha_i p^i (1-p)^{n-i} \quad .$$

Bsp. 3.6. Für den $[4,2]-code$ aus Bsp. 2.16. sind die Nebenklassenanführer $0000, 1000, 0100, 0010$. Daher gilt $\alpha_0 = 1, \quad \alpha_1 = 3, \quad \alpha_2 = \alpha_3 = \alpha_4 = 0$ und für $P_{corr}(c)$:

$$P_{corr}(c) = (1-p)^4 + 3p(1-p)^3$$

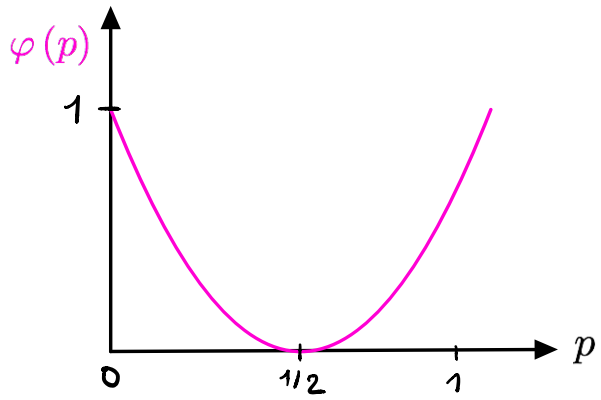
Bem. 3.7. Die Wahrscheinlichkeit, dass ein entschlüsseltes Wort nicht dem gesendeten Wort entspricht heißt Wort-Fehler-Rate und es gilt:

$$P_{err}(C) = 1 - P_{corr}(C)$$

Bem. 3.8. Ein linearer $[n,k]-code$ C nutzt n Zeichen um k Nachrichten zu senden. Man sagt dieser Code hat eine Rate von $R(c) = \frac{k}{n}$.
Ein „guter Code" wird eine hohe Rate haben.

Def. 3.9 Die Kapazität $\varphi(p)$ eines binären symmetrischen Kanals mit Zeichenfehlerwahrscheinlichkeit p ist

$$\varphi(p) = 1 + p\log(p) + (1-p)\log(1-p)$$



Theorem 3.10. Gegeben sei ein binärer symmetrischer Kanal mit Zeichenfehlerwahrscheinlichkeit p. Sei $R < \varphi(p)$. Dann:

$\forall \varepsilon > 0 \; \exists \, [n,k] - code \; C$ mit n hinreichend groß und $\frac{k}{n} > R$:

$P_{err}(C) < \varepsilon$.

Bsp. 3.11. Es gilt: $\varphi(0.01) \cong 0,92$. D.h. für $p = 0,01$ können wir selbst, wenn wir auf eine Rate von $\frac{9}{10}$ bestehen würden, theoretisch P_{err} so klein bekommen wie wir möchten, indem wir n und k hinreichend groß wählen.