INFORMATIONSTHEORIE

JKU MATHESEMINAR 2016

1. Ein typisches Problem

1.1. **Quellcodierung und Datenkompression.** Wir wollen eine Nachricht über einen digitalen Kanal, der nur 0 oder 1 übertragen kann, schicken. Die Nachricht ist eine Folge aus den Zeichen `A`, `B`, `C`, `D`, `E`. Eine typische Nachricht wäre etwa

```
AABAAAAAAAAACAAAAAAAABAAAAAAAAAEAAAAAAAAAAAEAAEAAEAAAAA
AAAEAACAABBAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
EAAAAAAAAAAAAAAEAAAAAABAAAAABAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAEAAEAAAAAAAAAAAAAAAAABAAAAAAAAAAAAAAAAAAAB
EAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAAAACAAAAAAAAAABAAAAAAB
AAAAAAAAAAAAAAAAAAAACABAAAABAAAAEAAAAAAAAAAAAABAAAAAAAA
CAAAABAAAAAAEAAAABAAAAAAAAAAAACAAAAAAAAAAAAAAAABAAAAAAA
BAAAAAACAAABAAAAAAAAAAAAAAAAAAAABAAAAAAAAAAAAAAAAAAAB
AAAAAAAAABAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABAAAABB
AAAAA
```

Dabei wissen wir, dass an jeder Stelle der Nachricht mit Wahrscheinlichkeit 0.9 ein `A`, mit 0.06 ein `B`, mit 0.015 ein `C`, mit 0.015 ein `D` und mit Wahrscheinlichkeit 0.01 ein `E` vorkommt.

Ziel ist es, für die Übertragung diese Nachricht als Folge von 0 und 1 zu kodieren. Dabei sollen wir für jedes Zeichen im Durchschnitt nur höchstens 0.8 Bits benötigen – eine Folge von 100 Zeichen sollte im Durchschnitt also auf 80 Bits komprimiert werden können.

Ist es möglich, eine solche Nachricht so stark zu komprimieren? Versuchen Sie, ein Verfahren zur Kompression zu finden!

Date: 14. Januar 2016.

Erhard Aichinger, Institut für Algebra, Johannes Kepler Universität Linz, `http://www.algebra.uni-linz.ac.at/`.

2. Einige Begriffe der Informationstheorie

Ein Kommunikationssystem besteht aus folgenden Teilen (s. [HQ95, Ash90]).

(1) Nachrichtenquelle
(2) Quellencodierer
(3) Kanalcodierer
(4) Kanal
(5) Rausch- oder Fehlerquelle
(6) Kanaldecodierer
(7) Quellendecodierer
(8) Nachrichtensenke

Die Begriffe “Verschlüsselung” und “Codierung”.

3. Wörter und eindeutig decodierbare Codes

Definition 3.1. Wörter über $\{0,1\}$, Zusammenhängen von Wörtern, $\{0,1\}^*$, Länge eines Wortes, $|W|$.

Definition 3.2. Sei $k \in \mathbb{N}$, und seien $p_1, p_2, \ldots, p_k \in [0,1]$ so, dass $\sum_{i=1}^{k} p_i = 1$, seien $A_1, \ldots, A_k \in \{0,1\}^*$, und sei $\mathcal{C} = (A_1, A_2, \ldots, A_k)$. Die *durchschnittliche Wortlänge $\overline{n}$ von $\mathcal{C}$* bezüglich $(p_1, \ldots, p_k)$ ist gegeben durch

$$\overline{n} = p_1|A_1| + p_2|A_2| + \cdots + p_k|A_k|.$$

Definition 3.3. Sei $k \in \mathbb{N}$. Ein *Code aus k Codewörtern* ist eine Folge $(A_1, \ldots, A_k)$ von Wörtern in $\{0,1\}^*$, sodass für alle $i, j \in \{1, \ldots, k\}$ mit $i \neq j$ auch $A_i \neq A_j$ gilt.

Definition 3.4. Sei $k \in \mathbb{N}$, und sei $\mathcal{C} = (A_1, \ldots, A_k)$ ein Code. $\mathcal{C}$ ist *eindeutig decodierbar*, wenn für alle $m, n \in \mathbb{N}$ und für alle $D_1, \ldots, D_m$ aus $\mathcal{C}$ und $E_1, \ldots, E_n$ aus $\mathcal{C}$ folgendes gilt: Wenn

$$D_1 * D_2 * \cdots * D_m = E_1 * E_2 * \cdots * E_n,$$

dann gilt $m = n$, und für alle $i \in \{1, \ldots, m\}$ gilt $D_i = E_i$.

Wie findet man eindeutig decodierbare Codes?

Übungsaufgaben 3.5.

(1) [HQ95, S.36] Zeigen Sie, dass der Code $\{0, 10, 011, 11111\}$ eindeutig decodierbar ist.
(2) Ist der Code $\{a, c, ad, abb, bad, deb, bbcde\}$ eindeutig decodierbar?
(3) Seien $V, W \in \{0,1\}^*$, also Wörter über dem Alphabet $\{0,1\}$. Wie müssen V und W aussehen, damit $V * W = W * V$ gilt? (Dabei ist $V * W$ das Wort, das durch Zusammenhängen von V und W entsteht.

4. Präfixfreie Codes

Definition 4.1. Seien A, B Wörter über $\{0,1\}$. A ist ein *Präfix* von B, wenn $A = B$, oder wenn es ein Wort C gibt, sodass $A * C = B$.

Definition 4.2. Seien $A_1, A_2, \ldots, A_k$ Wörter über $\{0,1\}$. $(A_1, A_2, \ldots, A_k)$ ist ein *präfixfreier Code*, wenn es keine $i, j \in \{1, 2, \ldots, k\}$ gibt, sodass $i \neq j$ und A_i ein Präfix von A_j ist.

Satz 4.3. *Präfixfreie Codes sind eindeutig decodierbar.*

Satz 4.4. *Sei $k \in \mathbb{N}$, und seien $n_1, n_2, \ldots, n_k \in \mathbb{N}$. Dann gibt es genau dann einen präfixfreien Code $(A_1, A_2, \ldots, A_k)$ über $\{0,1\}$, sodass $|A_1| = n_1$, …, $|A_2| = n_2$, $|A_k| = n_k$, wenn*

$$\frac{1}{2^{n_1}} + \frac{1}{2^{n_2}} + \cdots + \frac{1}{2^{n_k}} \leq 1.$$

Es gilt sogar für jeden eindeutig decodierbaren Code (nicht nur für jeden präfixfreien Code), dass

$$\frac{1}{2^{n_1}} + \frac{1}{2^{n_2}} + \cdots + \frac{1}{2^{n_k}} \leq 1.$$

Das ist aber schwieriger zu beweisen (cf. [Ash90, S. 35]).

Übungsaufgaben 4.5.

(1) Zeigen Sie: Präfixfreie Codes sind eindeutig decodierbar.
(2) Sei $k \in \mathbb{N}$, und seien $n_1, n_2, \ldots, n_k \in \mathbb{N}$. Zeigen Sie: Es gibt genau dann einen Präfixcode $(A_1, A_2, \ldots, A_k)$ über $\{0,1\}$, sodass $|A_1| = n_1, \ldots, |A_2| = n_2, |A_k| = n_k$, wenn

$$\frac{1}{2^{n_1}} + \frac{1}{2^{n_2}} + \cdots + \frac{1}{2^{n_k}} \leq 1.$$

5. Entropie

Definition 5.1. (Entropie) $H(p_1, \ldots, p_k) := \sum_{i=1}^{k} -p_i \log_2(p_i)$. Es gibt ein Problem, falls ein $p_i = 0$ ist.

Übungsaufgaben 5.2. Bestimmen Sie $H(0.5, 0.5)$, $H(0.3, 0.7)$, $H(0.1, 0.9)$, $H(.5, .25, .25)$, $H(0.6, 0.3, 0.05, 0.05)$.

Satz 5.3. *(Noiseless coding theorem) Sei $k \in \mathbb{N}$, und seien $p_1, p_2, \ldots, p_k \in (0, 1]$ so, dass $\sum_{i=1}^{k} p_i = 1$, sei $\mathcal{C} = (A_1, A_2, \ldots, A_k)$ ein Präfixcode, und sei $\overline{n}$ die durchschnittliche Wortlänge von $\mathcal{C}$ bezüglich $(p_1, \ldots, p_k)$. Dann gilt*

$$\overline{n} \geq H(p_1, \ldots, p_k).$$

Satz 5.4. *Sei $k \in \mathbb{N}$, seien $p_1, p_2, \ldots, p_k \in (0, 1]$ so, dass $\sum_{i=1}^{k} p_i = 1$, und sei $H := H(p_1, \ldots, p_k)$. Dann gibt es für diese Wahrscheinlichkeiten einen binären Präfixcode $(A_1, \ldots, A_k)$, für dessen durchschnittliche Wortlänge $\overline{n}$ bezüglich $(p_1, \ldots, p_k)$ gilt:*

$$H + 1 > \overline{n} \geq H.$$

Übungsaufgaben 5.5. In einer aus den drei Zeichen zusammengesetzten Nachricht kommen die Zeichen x_1, x_2, x_3 mit den Wahrscheinlichkeiten $0.9, 0.05, 0.05$ vor. Wir wollen diese Zeichen durch $0, 1$-Folgen übertragen.

(1) Man kann beweisen, dass wir das mit einem Code machen können, dessen durchschnittliche Codewortlänge im Intervall

$$[H(0.9, 0.05, 0.05), H(0.9, 0.05, 0.05) + 1]$$

liegt. Finden Sie einen solchen Code!

(2) Wir fassen jetzt immer zwei Nachrichtenzeichen zusammen und übertragen diese Ausgänge gemeinsam. Wieviele Zeichen brauchen Sie jetzt im Durchschnitt für die Übertragung eines Nachrichtenzeichens?

6. Konstruktion optimaler Codes

Sei $k \in \mathbb{N}$, und seien $p_1, \dots, p_k$ Zahlen in $(0, 1]$ mit $\sum_{i=1}^{k} p_i = 1$. Ein präfixfreier Code $\mathcal{C} = (C_1, C_2, \dots, C_k)$ ist *optimal* für $(p_1, \dots, p_k)$, wenn es keinen präfixfreien Code mit kleinerer durchschnittlicher Codewortlänge gibt.

Satz 6.1. *Sei $k \geq 3$, seien $p_1, \dots, p_k \in [0, 1]$ so, dass $\sum_{i=1}^{k} p_i = 1$ und $p_1 \geq p_2 \geq p_3 \geq \cdots \geq p_{k-1} \geq p_k \geq 0$, und sei $\mathcal{D} = (D_1, D_2, \dots, D_{k-1})$ ein optimaler Code für die Wahrscheinlichkeiten*

$$(p_1, p_2, \dots, p_{k-2}, p_{k-1} + p_k).$$

Sei $\mathcal{C} = (C_1, C_2, \dots, C_{k-1}, C_k)$ gegeben durch $C_i = D_i$ für $i \in \{1, 2, \dots, k-2\}$, $C_{k-1} = D_{k-1} * 0$, $C_k = D_{k-1} * 1$. Dann ist $\mathcal{C}$ ein optimaler Code für die Wahrscheinlichkeiten $(p_1, \dots, p_k)$.

Dieser Satz liefert den Huffman-Algorithmus.

Übungsaufgaben 6.2.

(1) Konstruieren Sie – etwa mit dem Verfahren von Huffman – optimale binäre Codes für folgende Wahrscheinlichkeiten, und berechnen Sie die durchschnittliche Codewortlänge.
   (a) $(0.6, 0.3, 0.05, .05)$
   (b) [Ash90] $(0.3, 0.25, 0.2, 0.1, 0.1, 0.05)$
   (c) $(0.2, 0.2, 0.2, 0.2, 0.2)$.
   (d) [Ash90] $(.2, .18, .1, .1, .1, .061, .059, .04, .04, .04, .04, .03, .01)$.
   (e) $(0.9, 0.06, 0.015, 0.015, 0.01)$.

(2) (cf. [CT06, Problem 5.25]) Seien $p_1 > p_2 \geq p_3 \geq p_4$ so, dass $p_1 + p_2 + p_3 + p_4 = 1$ und $p_1 > \frac{2}{5}$. Zeigen Sie, dass in einem mit dem Huffman-Algorithmus erzeugten optimalen binären Code (C_1, C_2, C_3, C_4) für (p_1, p_2, p_3, p_4) das Wort C_1 Länge 1 hat.

(3) Finden Sie $p_1 > p_2 \geq p_3 \geq p_4$ mit $p_1 = 0.39$, sodass in dem mit dem Huffman-Algorithmus erzeugten binären Code das Codewort für p_1 Länge 2 hat.

Buchstabenhäufigkeit (in Promille) in deutschen Texten ([Pom04]):

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
60	17	27	54	180	16	32	41	81	3	13	33	23	106	27	8	0	72	69	57	46	9	15	0	0	11

7. Überblick über die Informationstheorie

(Das ist das Kapitel 1 von [Aic07]).

7.1. **Einige Begriffe der Informationstheorie.** Ein Kommunikationssystem besteht aus folgenden Teilen (s. [HQ95, p. 1]).

(1) Nachrichtenquelle
(2) Quellencodierer
(3) Kanalcodierer
(4) Kanal
(5) Rausch- oder Fehlerquelle
(6) Kanaldecodierer
(7) Quellendecodierer
(8) Nachrichtensenke

7.2. **Quellencodierung und Datenkompression.** Wir stellen uns vor, wir haben eine Nachrichtenquelle, die jede Sekunde eines der Zeichen A, B, C, D liefert. Dabei sind 60% der Zeichen A, 30% der Zeichen B, 5% der Zeichen C und 5% der Zeichen D. Wir nehmen an, die Nachrichtenquelle produziert die Zeichen unabhängig voneinander; die Auswahl des nächsten Zeichens wird also von den bisher gelieferten Zeichen nicht beeinflusst.

Wir wollen eine von dieser Nachrichtenquelle produzierte Nachricht möglichst effizient als 0/1-Folge speichern.

Als erste Idee ersetzen wir jedes der 4 Zeichen durch eine 0/1 Folge. Wir vergleichen drei verschiedene Vorschläge.

Zeichen	Vorschlag 1	Vorschlag 2	Vorschlag 3
A	00	0	0
B	01	10	10
C	10	110	110
D	11	01	111

Wir überlegen uns, wie lange die 0/1-Folge ist, die wir brauchen, um eine Datei mit n Zeichen aus dieser Nachrichtenquelle zu speichern.

(1) Im Vorschlag 1 brauchen wir $2 \cdot n$ Zeichen für jede Datei mit n Zeichen. Wir brauchen also 2 Bits pro Nachrichtenzeichen.
(2) Der Vorschlag 2 hat folgenden Nachteil: DB und AC werden beide durch 0110 dargestellt. Daher ist ein eindeutiges Decodieren nicht mehr möglich.
(3) Im Vorschlag 3 ist ein eindeutiges Decodieren möglich. Hier werden verschiedene Dateien der Länge n zu möglicherweise unterschiedlich langen Dateien kodiert. Wenn wir eine "typische" Datei wählen, so hat diese Datei etwa $0.6\,n$ mal das Zeichen A, $0.3\,n$ mal das Zeichen B und jeweils

$0.05\,n$ mal die Zeichen C und D. Diese Datei wird mit

$$1 \cdot 0.6\,n + 2 \cdot 0.3\,n + 3 \cdot 0.05\,n + 3 \cdot 0.05\,n$$

$= 1.5\,n$ Bits dargestellt. Wir brauchen also 1.5 Bits/Nachrichtenzeichen.

Typische Resultate über die Quellcodierung:

(1) Shannon's Quellcodierungssatz (Noiseless Coding Theorem): Wenn die Zeichen $A_1, A_2, \ldots, A_n$ mit Wahrscheinlichkeiten $p_1, p_2, \ldots, p_n$ auftreten, so braucht jedes Quellcodierungsverfahren, das für beliebig lange Dateien funktioniert, im Mittel zumindest

$$\sum_{i=1}^{n} p_i \cdot \log_2(\frac{1}{p_i})$$

Bits pro Nachrichtenzeichen. Durch geeignete Codierungsverfahren kann man dieser Schranke beliebig nahe kommen. Für das obige Beispiel ist diese Schranke ungefähr 1.395 Bits pro Nachrichtenzeichen.

(2) Huffman-Algorithmus [Ash90, p. 42], [Mac03, p.99]: Für gegebene Zeichen $A_1, A_2, \ldots, A_n$ mit Wahrscheinlichkeiten $(p_1, p_2, \ldots, p_n)$ produziert der Huffman-Algorithmus die beste zeichenweise Codierung von $A_1, A_2, \ldots, A_n$ als 0/1-Folgen.

7.3. **Kanalcodierung.** Wir stellen uns vor, wir haben einen Kanal zur Verfügung, der 0 und 1 überträgt. Dabei kommt das Eingabezeichen 0 mit Wahrscheinlichkeit f (Fehlerwahrscheinlichkeit) als 1 und mit Wahrscheinlichkeit $1 - f$ als 0 an. Ebenso kommt ein gesandter 1er mit Wahrscheinlichkeit f als 0 und mit Wahrscheinlichkeit $1 - f$ als 1 an. Wir suchen nun Codierungsverfahren die so funktioneren:

(1) Eine Folge $x_1x_2x_3x_4\ldots$ von Bits soll übertragen werden.
(2) Der *Kanalcodierer* fügt Kontrollbits dazu und macht aus $x_1x_2x_3x_4$ die Folge $y_1y_2y_3y_4y_5y_6\ldots$.
(3) Diese Folge $y_1y_2y_3y_4y_5y_6\ldots$ wird über den Kanal gesendet. Die Folge $z_1z_2z_3z_4z_5z_6\ldots$ kommt an. Wir nehmen an, dass mit Wahrscheinlichkeit $1 - f$ ein z_i mit dem ausgesandten y_i übereinstimmt.
(4) Der *Kanaldecodierer* versucht, die vermutlich ausgesandte Folge $x_1x_2x_3x_4\ldots$ zu rekonstruieren. Er liefert also eine Bitfolge $u_1u_2u_3u_4\ldots$.

Die *Bitfehlerrate* ist die Wahrscheinlichkeit, dass ein Nachrichtenbit x_i nicht mit dem decodierten Bit u_i übereinstimmt.

Die *Übertragungsrate* ist die Anzahl der Nachrichtenbits pro gesendetem Kanalbit. Wir vergleichen zwei Varianten, um eine Bitfolge über einen Kanal zu schicken.

(1) "Keine" Codierung und Decodierung: Wir schicken die Nachricht ohne Kontrollstellen über den Kanal. Wenn f gegeben ist, können wir die Bitfehlerrate b und die Übertragungsrate r bestimmen:

$$b = f \text{ und } r = 1.$$

(2) Wir wiederholen jedes Bit drei Mal und decodieren durch Mehrheitsentscheidung. Es gilt:

$$b = f^3 + 3f^2(1-f) \text{ und } r = \frac{1}{3}.$$

Bei einem Kanal mit Fehlerwahrscheinlichkeit $f = 0.1$ können wir so eine Bitfehlerrate von 0.028 erreichen. Wenn wir durch den gleichen Kanal jedes Bit 1001 mal schicken und dann durch Mehrheitsentscheidung decodieren, erhalten wir die Fehlerwahrscheinlichkeit $b = 8.0276 \cdot 10^{-225}$. Der Preis, den wir dafür bezahlen, ist eine Ver-1001-fachung der Übertragungskosten, die Übertragungsrate ist in diesem Fall ja nur $r = \frac{1}{1001}$.

Wo liegen die theoretischen Grenzen für die Bitfehlerrate b bei gegebener Kanalfehlerrate f und Übertragungsrate r? Shannons Kanalcodierungssatz beantwortet diese Frage (cf. [Mac03, p. 15]).

(1) Wenn $r < 1 + f \cdot \log_2(f) + (1-f) \cdot \log_2(1-f)$ ist, und wenn ε eine vorgegebene maximal zulässige Bitfehlerrate mit $\varepsilon > 0$ ist, dann gibt es eine Codierung und Decodierung mit Übertragungsrate r, für die die Bitfehlerrate kleiner als ε ist.

    Der Preis, den wir für diese Sicherheit zahlen müssen, ist, dass wir eventuell lange Blöcke von N Nachrichtenzeichen abwarten müssen, um diese dann als $N \cdot \frac{1}{r}$ Bits zu kodieren. Dann erst senden wir diese $\frac{N}{r}$ Bits über den Kanal.

(2) Wenn $r > 1 + f \cdot \log_2(f) + (1-f) \cdot \log_2(1-f)$, dann ist die bestenfalls erreichbare Bitfehlerrate b jene Lösung der Gleichung

$$r \cdot \big(1 + b \cdot \log_2(b) + (1-b) \cdot \log_2(1-b)\big) = 1 + f \cdot \log_2(f) + (1-f) \cdot \log_2(1-f),$$

    für die $b < \frac{1}{2}$ ist. Kleinere Bitfehlerraten als dieses b sind unmöglich. Es gibt Codierungsverfahren (wieder mit Puffern der Nachricht), die diesem b beliebig nahe kommen.

Die Codierungstheorie bemüht sich, praktisch durchführbare Codierungsverfahren zu finden, die bei gegebenem r und f das b möglichst klein werden lassen (siehe [Wil99]).

Literatur

[Aic07] E. Aichinger, *Informations und Codierungstheorie (Informationstheorie)*, Lecture notes for a course at JKU Linz, Austria. Available at `www.algebra.uni-linz.ac.at/Students/InfoAndCoding/ws06/info-skriptum-v260107.pdf`, 2007.

[Ash90] R. B. Ash, *Information theory*, Dover Publications Inc., New York, 1990, Corrected reprint of the 1965 original.

[CT06] Thomas M. Cover and Joy A. Thomas, *Elements of information theory*, second ed., Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2006.

[HQ95] W. Heise and P. Quattrocchi, *Informations- und Codierungstheorie*, third ed., Springer-Verlag, Berlin, 1995.

[Mac03] D. J. C. MacKay, *Information theory, inference and learning algorithms*, Cambridge University Press, New York, 2003, The book can be viewed at `http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html`.

[Pom04] K. Pommerening, *Kryptologie (Vorlesungsunterlagen)*, Published electronically at `http://www.uni-mainz.de/~pommeren/Kryptologie/`, 2004.

[Wil99] W. Willems, *Codierungstheorie*, de Gruyter, Berlin, New York, 1999.