KAPITEL 1

Überblick

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

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.

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 $x_1 x_2 x_3 x_4 \ldots$ von Bits soll übertragen werden.
(2) Der *Kanalcodierer* fügt Kontrollbits dazu und macht aus $x_1 x_2 x_3 x_4$ die Folge $y_1 y_2 y_3 y_4 y_5 y_6 \ldots$.

(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**]).

Literaturverzeichnis

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

[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`.

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