

Quellencodierung I: Redundanzreduktion, redundanzsparende Codes

1. Quellencodierung

Durch die *Quellencodierung* werden die Daten aus der Quelle codiert, bevor sie in einem Übertragungskanal übertragen werden.

Die Codierung dient der Verkleinerung der zu übertragenden Datenmenge (Kompression).

2. Redundanz

2.1 Einführung

Redundanz: Bezeichnung für die Anteile einer Nachricht, die keine Information vermitteln, also überflüssig sind... .

(aus: Duden Informatik, Dudenverlag, Mannheim 1993)

Beispiel: BCD-Code

Zeichen	Codewort
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Der Code ist redundant, weil nur 10 der 16 möglichen Codewörter benutzt werden.

*

Beispiel: ASCII-Code

Zeichen	Codewort
NUL	0000000
SOH	0000001
STX	0000010
...	...
x	1111000
y	1111001
z	1111010
{	1111011
	1111100
}	1111101
~	1111110
DEL	1111111

Alle 128 möglichen Codewörter werden genutzt (Minimalcode).
Trotzdem ist der Code redundant, denn die Zeichen treten mit unterschiedlichen Wahrscheinlichkeiten auf.

*

2.2 quantitative Erfassung der Redundanz

Definition: Die **Codewortlänge** eines Codewortes ist die Anzahl der Binärzeichen (0 oder 1), die bei der Codierung des Codewortes benutzt werden.

Definition: Die **mittlere Codewortlänge** m ist der Mittelwert (d.h. der Erwartungswert) der Codewortlängen aller n Zeichen:

$$m := \sum_{i=1}^n p(x_i) \cdot m_i$$

mit $p(x_i)$: Wahrscheinlichkeit, daß das i -te Zeichen auftritt,

m_i : Codewortlänge des i -ten Zeichens.

Definition: Der **mittlere Informationsgehalt** H von n Zeichen ist der Mittelwert des Informationsgehaltes I_i der einzelnen Zeichen:

$$H := \sum_{i=1}^n p(x_i) \cdot I_i$$

mit $I_i = \log_2 \left(\frac{1}{p(x_i)} \right)$: Informationsgehalt des i -ten Zeichens,

$p(x_i)$: Wahrscheinlichkeit, daß das i -te Zeichen auftritt.

Definition: Die **Redundanz** eines Codes ist die Differenz zwischen der mittleren Codewortlänge und dem mittleren Informationsgehalt:

$$R := m - H.$$

Die **relative Redundanz** gibt an, wieviel Prozent der Codierung redundant sind:

$$r := \frac{m - H}{m}.$$

2.3 Redundanzreduktion

Die Redundanz wird reduziert, indem statt Blockcodes Codewörter unterschiedlicher Länge benutzt werden.

Zeichen, die mit hoher Wahrscheinlichkeit auftreten, erhalten ein kurzes Codewort, Zeichen, die selten auftreten, ein langes.

Bei der Codierung wird die mittlere Codewortlänge kleiner als die Länge eines Blockcodes.

Problem: Die Länge der Codewörter muß übermittelt werden, damit die Codewörter wieder eindeutig decodiert werden können.

Beispiel:

Zeichen	Codewort
1	0
2	1
3	10

Die Zeichenfolge 1, 2, 3 wird codiert als 0110. Die Decodierung ist nicht eindeutig: 0110 kann entweder 1, 2, 3 oder 1, 2, 2, 1 bedeuten.

*

Fano-Bedingung (Präfix-Eigenschaft):

Kein Codewort aus einem Code bildet den Anfang eines anderen Codewortes.

Wenn diese Bedingung erfüllt ist, muß die Codewortlänge nicht übertragen werden, und die Decodierung ist eindeutig.

Beispiel:

Telefonnummern sind nach der Fano-Bedingung codiert.

*

Shannonsches Codierungstheorem:

Die Codierung eines Zeichenvorrats kann immer so vorgenommen werden, daß die Redundanz minimal wird.

3. redundanzsparende Codes

3.1 Codierung nach Shannon

Voraussetzungen:

- Die n verschiedenen Zeichen x_1, x_2, \dots, x_n treten mit den Wahrscheinlichkeiten $p(x_1), p(x_2), \dots, p(x_n)$ auf.
- Die Zeichen sind nach fallender Wahrscheinlichkeit geordnet:
 $p(x_1) \geq p(x_2) \geq \dots \geq p(x_n)$.
- Für jedes Zeichen x_i ist P_i die kumulative Wahrscheinlichkeit aller vorigen Zeichen. Das bedeutet

$$P_1 = 0,$$

$$P_2 = p(x_1),$$

$$P_3 = p(x_1) + p(x_2),$$

$$P_4 = p(x_1) + p(x_2) + p(x_3),$$

...

$$P_n = p(x_1) + p(x_2) + \dots + p(x_{n-1}).$$

Allgemein: $P_{i+1} = P_i + p(x_i)$ für $i = 1, \dots, n-1$.

Berechnung der Codewortlänge:

Die zu benutzende Codewortlänge des i -ten Zeichens wird berechnet als

$$m_i = \lceil I_i \rceil = \left\lceil \log_2 \left(\frac{1}{p(x_i)} \right) \right\rceil.$$

Sie ist also die kleinste natürliche Zahl, die größer oder gleich dem Informationsgehalt des Zeichens x_i ist.

Berechnung der Codewörter:

Die kumulative Wahrscheinlichkeit P_i , die zum Zeichen x_i gehört, wird in eine Dualzahl umgerechnet.

Dabei gelten folgende Regeln:

- Die Dualzahl muß möglichst groß, aber kleiner als P_i sein.
- Die Vorkommatstelle wird ignoriert, weil sie sowieso immer 0 ist (sozusagen ein „hidden bit“); es werden also nur die Nachkommastellen umgerechnet.
- Die Dualzahl wird das gesuchte Codewort.
- Die Dualzahl wird nach der m_i -ten Stelle abgebrochen, denn das Codewort soll genau m_i Stellen lang sein.

Man erhält also die Binärzeichen b_1, b_2, \dots, b_{m_i} , die das gesuchte Codewort ergeben.

Beispiel:

Zeichen x_i	$p(x_i)$	P_i	m_i	Codewort	Z_i
A	0,4	0	2	00	0
B	0,2	0,4	3	011	0,375
C	0,15	0,6	3	100	0,5
D	0,15	0,75	3	110	0,75
E	0,05	0,9	5	11100	0,875
F	0,05	0,95	5	11110	0,9375

Die mittlere Codewortlänge ist $m=2,8$ Bit; der mittlere Informationsgehalt ist $H=2,246$ Bit. Daraus ergeben sich die Redundanz $R=0,554$ Bit und die relative Redundanz $r=19,77\%$.

*

Beweis der Präfixeigenschaft:

Hilfssatz 1: Die kumulative Wahrscheinlichkeit eines Zeichens und die entsprechende Dualzahl unterscheiden sich in den ersten m_i Stellen nicht, d.h. $P_i - Z_i < 2^{-m_i}$.

Beweis: Die nicht abgebrochene Dezimaldarstellung von P_i sei

$$P_i = a_1 \cdot 2^{-1} + a_2 \cdot 2^{-2} + \dots + a_{m_i} \cdot 2^{-m_i} + a_{m_i+1} \cdot 2^{-m_i-1} + a_{m_i+2} \cdot 2^{-m_i-2} + \dots \text{ (für } a_i \in \{0,1\} \text{)}.$$

Die Zahl Z_i , die durch Umrechnen von P_i nach den Regeln in eine Dualzahl entstanden ist, sei $Z_i = b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \dots + b_{m_i} \cdot 2^{-m_i}$. Die Zahl Z_i ist also eine Näherung an P_i .

Die Differenz beider Zahlen ist dann

$$P_i - Z_i = (a_1 - b_1) \cdot 2^{-1} + (a_2 - b_2) \cdot 2^{-2} + \dots + (a_{m_i} - b_{m_i}) \cdot 2^{-m_i} + a_{m_i+1} \cdot 2^{-m_i-1} + \dots$$

Nach der Konstruktionsregel muß die Dualzahl kleiner als P_i sein, d.h. $Z_i \leq P_i$. Weil

die Zahl Z_i aber auch möglichst groß sein soll, muß $a_1 = b_1 \wedge a_2 = b_2 \wedge \dots \wedge a_{m_i} = b_{m_i}$ gelten. Denn wenn dies nicht gilt, gibt es entweder noch eine größere Zahl, die $Z_i \leq P_i$ erfüllt, oder die Bedingung $Z_i \leq P_i$ wurde verletzt.

Dazu ein Beispiel: $P_i = 001100100$, $Z_i = 001100$. Wenn ein Bit in Z_i auf 1 statt auf 0 ist, dann ist $Z_i \leq P_i$ verletzt; wenn ein Bit in Z_i auf 0 statt auf 1 ist, dann gibt es eine größere Zahl, nämlich die, die oben steht.

Es folgt also $P_i - Z_i = a_{m_i+1} \cdot 2^{-m_i-1} + \dots$, was sich abschätzen läßt als $P_i - Z_i < 2^{-m_i}$.

Hilfssatz 2: Die Differenz zwischen zwei aufeinander folgenden kumulativen Wahrscheinlichkeiten ist mindestens 2^{-m_i} , d.h. $P_{i+1} - P_i \geq 2^{-m_i}$.

Beweis: Es ist $m_i = \left\lceil \log_2 \left(\frac{1}{p(x_i)} \right) \right\rceil$ die Codewortlänge. Daraus folgt $m_i \geq \log_2 \left(\frac{1}{p(x_i)} \right)$.

$$\text{Umformung nach } p(x_i) \text{ ergibt: } m_i \geq \log_2 \left(\frac{1}{p(x_i)} \right) \Leftrightarrow 2^{m_i} \geq \frac{1}{p(x_i)} \Leftrightarrow p(x_i) \geq \frac{1}{2^{m_i}}$$

$$\Leftrightarrow p(x_i) \geq 2^{-m_i}.$$

Weil man die kumulative Wahrscheinlichkeit aus $P_{i+1} = P_i + p(x_i)$ berechnen kann, folgt

$$P_{i+1} - P_i \geq 2^{-m_i}.$$

Mit den beiden Hilfssätzen wird nun die Präfixeigenschaft des Shannon-Codes bewiesen. Die Codeeigenschaft $P_i - Z_i < 2^{-m_i}$ bedeutet, daß sich die kumulative Wahrscheinlichkeit des vorigen Zeichens und die entsprechende Dualzahl in keiner Stelle unterscheiden. Anders gesagt, gibt die kumulative Wahrscheinlichkeit P_i als Dualzahl gerade das entsprechende Codewort an.

Die Aussage $P_{i+1} - P_i \geq 2^{-m_i}$ besagt, daß *aufeinanderfolgende* Codewörter von der ersten bis zur m_i -ten Stelle verschieden sein müssen. Das ist die Präfixeigenschaft, für zwei aufeinanderfolgende Codewörter gilt sie also.

Die Formel $P_{i+1} - P_i \geq 2^{-m_i}$ gilt für beliebige i . Wenn man beispielsweise für i einfach $i+1$ einsetzt, erhält man $P_{i+2} - P_{i+1} \geq 2^{-m_{i+1}}$. Diese Gleichung zur ursprünglichen Formel addiert, ergibt $P_{i+2} - P_i + P_{i+1} - P_{i+1} \geq 2^{-m_i} + 2^{-m_{i+1}}$, woraus folgt $P_{i+2} - P_i \geq 2^{-m_i} + 2^{-m_{i+1}}$. Das bedeutet, daß auch für Codewörter im Abstand 2 die Präfixeigenschaft erfüllt ist, weil sie in den ersten m_i Stellen verschieden sein müssen.

Die Präfixeigenschaft zweier beliebiger Codewörter im Abstand n ergibt sich dementsprechend dadurch, daß man in der Formel $P_{i+1} - P_i \geq 2^{-m_i}$ die Variable i jeweils durch $i+1, i+2, \dots, i+n-1$ ersetzt und alle Gleichungen, die man dabei erhält, addiert.

3.2 Codierung nach Fano

Voraussetzungen:

- Die n verschiedenen Zeichen x_1, x_2, \dots, x_n treten mit den Wahrscheinlichkeiten $p(x_1), p(x_2), \dots, p(x_n)$ auf.
- Die Zeichen sind nach fallender Wahrscheinlichkeit geordnet:
 $p(x_1) \geq p(x_2) \geq \dots \geq p(x_n)$.

Codierungsvorschrift:

1. Die Menge der Zeichen wird in 2 Teile aufgeteilt, die möglichst gleichwahrscheinlich sind.
2. Der einen Teilmenge wird die Null als erste Codewortstelle zugeordnet, der anderen die Eins.
3. Für beide Teilmengen werden diese Schritte rekursiv durchgeführt, bis Teilmengen entstehen, die nur ein Zeichen enthalten.

Beispiel:

Zeichen x_i	$p(x_i)$	Codewort
A	0,4	00
B	0,2	01
C	0,15	10
D	0,15	110
E	0,05	1110
F	0,05	1111

Die mittlere Codewortlänge ist $m=2,35$ Bit; der mittlere Informationsgehalt ist $H=2,246$ Bit. Daraus ergeben sich die Redundanz $R=0,104$ Bit und die relative Redundanz $r=4,41\%$.

*

3.3 Codierung nach Huffman

Voraussetzungen:

- Die n verschiedenen Zeichen x_1, x_2, \dots, x_n treten mit den Wahrscheinlichkeiten $p(x_1), p(x_2), \dots, p(x_n)$ auf.
- Die Zeichen sind nach fallender Wahrscheinlichkeit geordnet:
 $p(x_1) \geq p(x_2) \geq \dots \geq p(x_n)$.

Codierungsvorschrift:

1. Zwei Zeichen der kleinsten Wahrscheinlichkeit werden herausgesucht.
2. Dem einen Zeichen wird die Null als (links anzufügende) Codewortstelle zugeordnet, dem anderen die Eins.
3. Die beiden Zeichen werden zu einem einzigen zusammengefaßt (Addition der Wahrscheinlichkeiten).
4. Das Verfahren wird wiederholt, bis alle Zeichen zusammengefaßt sind.

Die Huffman-Codes führen zu den kleinstmöglichen mittleren Codewortlängen.

Beispiel:

Schritt 1:

Zeichen x_i	$p(x_i)$	Codestelle
A	0,4	
B	0,2	
C	0,15	
D	0,15	
E	0,05	0
F	0,05	1

Schritt 2:

Zeichen x_i	$p(x_i)$	Codestelle
A	0,4	
B	0,2	
C	0,15	
D	0,15	0
E, F	0,1	1

Schritt 3:

Zeichen x_i	$p(x_i)$	Codestelle
A	0,4	
D, E, F	0,25	
B	0,2	0
C	0,15	1

Schritt 4:

Zeichen x_i	$p(x_i)$	Codestelle
A	0,4	
B, C	0,35	0
D, E, F	0,25	1

Schritt 5:

Zeichen x_i	$p(x_i)$	Codestelle
A	0,4	0
B, C, D, E, F	0,6	1

Zeichen x_i	$p(x_i)$	Codewort
A	0,4	0
B	0,2	100
C	0,15	101
D	0,15	110
E	0,05	1110
F	0,05	1111

Die mittlere Codewortlänge ist $m=2,3$ Bit; der mittlere Informationsgehalt ist $H=2,246$ Bit. Daraus ergeben sich die Redundanz $R=0,054$ Bit und die relative Redundanz $r=2,33\%$.

*

3.3 Beweis Shannonsches Codierungstheorem

Hilfssatz: Beim Shannon-Code kann man die mittlere Codewortlänge abschätzen als $H \leq m < 1 + H$. Das bedeutet, daß die mittlere Codewortlänge m den mittleren Informationsgehalt um weniger als 1 Bit übersteigt.

Beweis: Die Codewortlänge beim Shannon-Code ist $m_i = \left\lceil \log_2 \left(\frac{1}{p(x_i)} \right) \right\rceil$. Daraus folgt

einerseits $m_i \geq \log_2 \left(\frac{1}{p(x_i)} \right)$ und andererseits $m_i < 1 + \log_2 \left(\frac{1}{p(x_i)} \right)$.

Die mittlere Codewortlänge m läßt sich damit auf zwei Arten abschätzen:

$$m = \sum_{i=1}^n p(x_i) \cdot m_i \geq \sum_{i=1}^n p(x_i) \cdot \log_2 \left(\frac{1}{p(x_i)} \right) = H \quad \text{und als zweites}$$

$$m = \sum_{i=1}^n p(x_i) \cdot m_i < \sum_{i=1}^n p(x_i) \cdot \left(1 + \log_2 \left(\frac{1}{p(x_i)} \right) \right) = \sum_{i=1}^n \left(p(x_i) + p(x_i) \cdot \log_2 \left(\frac{1}{p(x_i)} \right) \right)$$

$$= \sum_{i=1}^n p(x_i) + \sum_{i=1}^n p(x_i) \cdot \log_2 \left(\frac{1}{p(x_i)} \right) = 1 + H.$$

Man erhält also $H \leq m < 1 + H$.

Beweis des Shannonschen Codierungstheorems:

Der Shannon-Code bietet eine beliebige Annäherung an einen idealen Code.

Wir betrachten nun n verschiedene Zeichen, die zu Zeichenfolgen der Länge N zusammengesetzt werden. Alle möglichen Zeichenfolgen werden mit dem Shannon-Code codiert (es gibt n^N verschiedene mögliche Zeichenfolgen).

Jedes Zeichen der Zeichenfolge habe den mittleren Informationsgehalt H ; der mittlere Informationsgehalt der Zeichenfolge ist dann $H_N = H \cdot N$.

Für die Zeichenfolge gilt die obige Abschätzung für die mittlere Codewortlänge

$H_N \leq m < 1 + H_N$, also folgt $H \cdot N \leq m < 1 + H \cdot N$. Wenn man die mittlere Codewortlänge nun nicht auf die Zeichenfolge, sondern auf die einzelnen Zeichen bezieht, ergibt sich

$$H \leq m' < \frac{1}{N} + H \quad (m' \text{ ist die mittlere Codewortlänge eines codierten Zeichens: } m' = \frac{m}{N}).$$

$$\text{Daraus folgt } 1 < \frac{1}{N \cdot m'} + \frac{H}{m'}$$

Damit läßt sich die relative Redundanz abschätzen:

$$r = \frac{m' - H}{m'} = 1 - \frac{H}{m'} < \frac{1}{N \cdot m'} + \frac{H}{m'} - \frac{H}{m'} = \frac{1}{N \cdot m'}. \text{ Diese Ungleichung läßt sich weiter}$$

$$\text{abschätzen: Wegen } H \leq m' \Leftrightarrow \frac{1}{m'} \leq \frac{1}{H} \Leftrightarrow \frac{1}{N \cdot m'} \leq \frac{1}{N \cdot H} \text{ folgt nämlich } r < \frac{1}{N \cdot m'} \leq \frac{1}{N \cdot H}$$

(es gilt sogar $r \leq \frac{1}{N \cdot H}$). Für wachsendes N strebt die relative Redundanz gegen Null, denn

$$\text{aus } 0 \leq r \leq \lim_{N \rightarrow \infty} \frac{1}{N \cdot H} = 0 \text{ folgt } r = 0.$$

Das bedeutet, daß bei der Codierung langer Zeichenfolgen die Redundanz gegen Null strebt.

3.4 Codeumschaltung

Bei der Codeumschaltung werden zwei Codetabelle benutzt, zwischen denen mittels spezieller Codewörter umgeschaltet werden kann.
Damit werden statistische Abhängigkeiten ausgenutzt, um die Redundanz zu verringern.

Beispiel: Telegraphenalphabet

Buchstaben	Ziffern	Codezeichen
A	-	11000
B	?	10011
C	:	01110
D	wer da?	10010
E	3	10000
F	frei	10110
G	frei	01011
H	frei	00101
I	8	01100
J	BEL	11010
K	(11110
L)	01011
M	.	00111
N	,	00110
O	9	00011
P	0	01101
Q	1	11101
R	4	01010
S	‘	10100
T	5	00001
U	7	11100
V	=	01111
V	2	11001
X	/	10111
Y	6	10101
Z	+	10001
CR	CR	00010
LF	LF	01000
Buchstaben	Buchstaben	11111
Ziffern	Ziffern	11011
SP	SP	00100
		00000

*

Quellen: T. Grams: Codierungsverfahren, BI-Wissenschaftsverlag, 1986
Bärbel Mertsching: Grundzüge der Informatik B1, Uni HH, 1997