ETH Zürich
Dept. Informatik

Prof. Markus Gross
WS 05/06

Informationstheorie

Lösung 8

8.1 Shannon-Fano Codierung

a) Das Applet liefert folgenden Code:

	a	b	c	d	e	f	g	$E[l_c]$
Code	001	011	11	10	0000	0001	010	2.63

b) Beide Zeichen werden mit nur einem Bit codiert, deshalb ist der Erwartungswert für die Codewortlänge $E[l_c] = 1$.

Die Entropie ist

$$H(X) = -0.1 * log_2(0.1) - 0.9 * log_2(0.9) = 0.469.$$

Die Coderedundanz ist demzufolge

$$R_C = E[l_c] - H(X) = 1 - 0.469 = 0.531.$$

c) Die erweiterte Quelle hat vier verschiedene Zeichen: *aa,ab,ba,bb*. Die Verteilung sieht folgendermassen aus:

X_2	$P(X_2)$	Code
aa	0.1*0.1 = 0.01	000
ab	0.1*0.9 = 0.09	001
ba	0.9*0.1 = 0.09	01
bb	0.9*0.9 = 0.81	1

Daraus lässt sich der Erwartungswert für die Codewortlänge berechnen:

$$E[l_c] = \frac{3*0.01+3*0.09+2*0.09+1*0.81}{2} = 0.645$$

Wichtig ist hier zu beachten, dass durch zwei geteilt wird, da wir den Erwartungswert für das Ursprungsalphabet erhalten möchten und wir in der erweiterten Quelle immer zwei Zeichen zusammengefasst haben.

Die Entropie ist immer noch gleich wie vorher

$$H(X) = -0.1 * log_2(0.1) - 0.9 * log_2(0.9) = 0.469.$$

Daraus folgt für die Coderedundanz:

$$R_C = E[l_c] - H(X) = 0.645 - 0.469 = 0.176$$

Die Coderedundanz ist im Vergleich zur vorherigen Aufgabe schon stark gesunken. Würden wir noch mehr als zwei Zeichen zusammenfassen, so würde die Coderedundanz noch weiter gegen 0 sinken.

8.2 Arithmetische Codierung

a) Die Codierung geht wie folgt vor:

i. Teile das Intervall $[0, 1)$ in Subintervalle gemäss der Symbol-Wahrscheinlichkeiten. Also hier $[0, 0.5)$, $[0.5, 0.8)$ und $[0.8, 1)$.

ii. Nimm das nächste Symbol aus dem String und ermittle das Subintervall, welches seiner Wahrscheinlichkeit entspricht. Wenn das nächste Symbol ein a ist, dann nehmen wir das Intervall $[0, 0.5)$.

iii. Teile dieses Intervall gemäss der Symbolwahrscheinlichkeiten, so wie in Schritt 1. $[0, 0.5)$ wird in $[0, 0.25)$, $[0.25, 0.4)$ und $[0.4, 0.5)$ unterteilt.

iv. Führe die Schritte 2 und 3 aus, bis der String zu Ende ist. Für den String *aacbca* erhalten wir das Intervall $[0.237, 0.2385)$.

v. Jede Zahl im Intervall $[0.237, 0.2385)$ ist ein gültiges Codewort für die gegebene Zeichenfolge. Weise dem String eine Zahl aus diesem Intervall zu. Nehmen wir zum Beispiel den Mittelpunkt des Intervalls, dann bekommen wir 0.23775 für *aacbca*.

b) Um eine Zahl zu decodieren gehen wir ähnlich vor:

i. Teile das Intervall $[0, 1)$ in Subintervalle gemäss der Symbol-Wahrscheinlichkeiten. Also hier $[0, 0.5)$, $[0.5, 0.8)$ und $[0.8, 1)$.

ii. Nimm das Intervall zu welchem die Zahl gehört und ermittle das entsprechende Symbol. Die Zahl 0.633 hat als erstes Symbol b.

iii. Teile dieses Intervall gemäss der Symbolwahrscheinlichkeiten, so wie in Schritt 1. $[0.5, 0.8)$ wird in $[0.5, 0.65)$, $[0.65, 0.74)$ und $[0.74, 0.8)$ geteilt.

iv. Führe die Schritte 2 und 3 so oft aus, bis die Länge des Wortes erreicht ist. Für die Zahl 0.633 erhalten wir den String *bacac*.

c) Wir müssen zeigen, dass $\lfloor \overline{T}(x) \rfloor_{l(x)} \in \left[u^{(m)}, o^{(m)}\right)$.

Beweis von $\lfloor \overline{T}(x) \rfloor_{l(x)} < o^{(m)}$:
Da $\overline{T}(x)$ der Mittelwert von $o^{(m)}$ und $u^{(m)}$ ist und gleichzeitig $o^{(m)} \neq u^{(m)}$ gelten muss, da sonst das Intervall und somit die Wahrscheinlichkeit dieses Codewortes 0 sein würde, gilt $\overline{T}(x) < o^{(m)}$. Weil das Abschneiden des Bitstrings die Zahl nur kleiner machen kann und nicht grösser, gilt $\lfloor \overline{T}(x) \rfloor_{l(x)} \leq \overline{T}(x)$. Daraus folgt $\lfloor \overline{T}(x) \rfloor_{l(x)} < o^{(m)}$.

Beweis von $u^{(m)} \leq \lfloor \overline{T}(x) \rfloor_{l(x)}$:
Um die zweite Ungleichung zu zeigen, betrachten wir

$$\begin{aligned}
\overline{T}(x) - \lfloor \overline{T}(x) \rfloor_{l(x)} &< 2^{-l(x)} \\
&= \frac{1}{2^{\lceil \log \frac{1}{P(x)} \rceil + 1}} \\
&\leq \frac{1}{2^{\log \frac{1}{P(x)} + 1}} \\
&= \frac{1}{2\frac{1}{P(x)}} = \frac{P(x)}{2}.
\end{aligned}$$

Es ergibt sich also

$$\lfloor \overline{T}(x) \rfloor_{l(x)} > \overline{T}(x) - \tfrac{P(x)}{2} = u^{(m)}.$$

Die letzte Gleichung gilt, da $P(x)$ die Wahrscheinlichkeit von x ist, gleichzeitig aber per Definition auch die Länge des Intervalls bezeichnet.

8.3 Codierung der ganzen Zahlen

a) Die beiden Teile der Codewörter in C_1 und C_2 sind jeweils mit | getrennt.

j	$B(j)$	$L(j)$	$C_1(j)$	$C_2(j)$
1	1	1	|1	1|
2	10	2	0|10	010|0
3	11	2	0|11	010|1
4	100	3	00|100	011|00
5	101	3	00|101	011|01
6	110	3	00|110	011|10
7	111	3	00|111	011|11
8	1000	4	000|1000	00100|000
9	1001	4	000|1001	00100|001
10	1010	4	000|1010	00100|010

$C_1(2^{19}) = 00000000000000000|10000000000000000000$ (Länge 39), und
$C_2(2^{19}) = 000010100|000000000000000000$ (Länge 28): für grosse Zahlen ist C_2 kürzer als C_1!

b) Weil $C_1(0)$ nicht definiert ist.

c) Für C_2' werden die ersten drei Codewörter abgeändert: $C_2'(1) = 10$, $C_2'(2) = 110$, $C_2'(3) = 111$. Es gilt:

$$E[l_{C_2}(X)] = P(1) + 4P(2) + 4P(3) + \sum_{x>3} P(x) l_{C_2}(x),$$

$$E[l_{C_2'}(X)] = 2P(1) + 3P(2) + 3P(3) + \sum_{x>3} P(x) l_{C_2'}(x).$$

Dann ist für alle Wahrscheinlichkeitsverteilungen mit $P(1) < P(2) + P(3)$:

$$\begin{aligned}
E[l_{C_2}(X)] - E[l_{C_2'}(X)] &= (P(1) + 4P(2) + 4P(3)) \; - \; (2P(1) + 3P(2) + 3P(3)) \\
&= -P(1) + P(2) + P(3) \\
&> 0.
\end{aligned}$$