Prof. Dr. Stefan Weinzierl 19.12.2013

Musterlösung: 19. Dezember 2013, 08:34

1 Fehlerkorrektur II

Audiosymbole mit einer Länge von 8 bit werden mit einem Paritätsbit zur Fehlererkennung kodiert.

a) Wird bei einer Paritätsprüfung das empfangene Kodewort 100100101 als fehlerfrei klassifiziert?
Lösung: Je nach Paritäts-Typ würde dieses Wort als gültig oder ungültig erkannt werden. Bei der häufiger anzutreffenden „even parity" wird das angefügte Prüfbit so gewählt, dass die Anzahl der Einsen im Gesamtwort (9-bit) gerade ist. Für diesen Fall wäre unser Kodewort gültig, auch wenn wir damit die (geringe) Restfehlerwahrscheinlichkeit, dass genau 2, 4, 6 oder gar 8 Bit fehlerhaft sind, außer Acht lassen. Bei „odd parity" würde das Kodewort als fehlerhaft erkannt werden. JA für „even parity" NEIN für „odd parity" Ein Nachteil einfacher Paritätskodes ist, dass nur eine ungeradzahlige Anzahl von fehler-haften Bits erkannt wird. Zudem können diese nicht korrigiert werden, da durch den Paritätscheck nur erkannt wird, dass ein Fehler existiert, nicht aber wo er sich befindet.

b) Wie groß ist die Wahrscheinlichkeit einer falschen Klassifizierung, wenn in dem so kodierten Kanal Bitfehler (random bit errors) mit einer Bit Error Rate (BER) von 10^{-3} auftreten?
Lösung: Die Wahrscheinlichkeit einer falschen Klassifizierung, d.h. der fälschlichen Akzeptanz eines fehlerhaften Kodewortes durch die Paritätsprüfung kann man mithilfe der Bernoullischen Formel zur Binominalverteilung berechnen.

$$p_k = \binom{n}{k} p^k \cdot (1-p)^{n-k} \qquad , mit \qquad \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Sie gibt die Wahrscheinlichkeit an, dass bei einem n mal wiederholten Zufallsexperiment (n voneinander unabhängigen Einzelversuche) das Ereignis A mit der Wahrscheinlichkeit p genau k mal auftritt. Im Kontext der Bitfehlerbetrachtung ist dabei

- A: Auftreten eines fehlerhaften Bits
- $n = 9$ Bit bzw. 9 Versuche je Kodewort
- k: Anzahl fehlerhafter Bits (Ereignisse) pro Kodewort (Versuchsreihe n)
- p: Auftretenswahrscheinlichkeit eines Bitfehlers

Der Term p^k beschreibt die Wahrscheinlichkeit, dass genau k Bits fehlerhaft sind (multiplikative Verknüpfung der Wahrscheinlichkeiten konjunkter Ereignisse: $p_{tot} = p(A)p(B)$). Der Term $(1-p)^{n-k}$ beschreibt die Wahrscheinlichkeit, dass die restlichen $n-k$ Bits fehlerfrei sind. Die Konjunktivität (Gleichzeitigkeit) der zwei beschriebenen Zustände verlangt die multiplikative Zusammenfassung der zwei Terme. Der Term $\binom{n}{k}$ beschreibt die Gesamtanzahl aller verschiedenen Anordnungen von k fehlerhaften Bits in einem n-stelligen Kodewort (Fehlermuster) und muss mit den beiden anderen Termen multipliziert werden.
Um nun die Wahrscheinlichkeit für Falschklassifikation bei Paritätskanalcodierung zu bestimmen, muss man sich zuerst klarmachen, welche Fehler zur Falschakzeptanz eines Kodeworts führen. Die

Paritätsprüfung erkennt dann Fehler nicht, wenn sie geradzahlig d.h. 2-, 4-, 6-, 8-fach usw. auftreten. Die Kodewortfehlerwahrscheinlichkeit ergibt sich nach dem Additionstheorem für disjunkte Ereignisse ($p_{tot} = p(A) + p(B)$) als Summe der Einzelwahrscheinlichkeiten:

$$p_{Kodewortfehler} = p_{2\ Fehler} + p_{4\ Fehler} + p_{6\ Fehler} + p_{8\ Fehler}$$

Man berechnet also die Auftretenswahrscheinlichkeiten p aller Bitfehlermuster, die von der Paritätsprüfung nicht erkannt werden und summiert diese zur Gesamtfehlerwahrscheinlichkeit auf.

$$\begin{aligned}
p_{Kodewortfehler} &= \sum_k p_k \rightarrow k = \{2,\ 4,\ 6,\ 8\} \\
p_{Kodewortfehler} &= \binom{9}{2} 10^{-3 \cdot 2} \left(1 - 10^{-3}\right)^7 + \binom{9}{4} 10^{-3 \cdot 4} \left(1 - 10^{-3}\right)^5 \\
&\quad + \binom{9}{6} 10^{-3 \cdot 6} \left(1 - 10^{-3}\right)^3 + \binom{9}{8} 10^{-3 \cdot 8} \left(1 - 10^{-3}\right)^7 \\
p_{Kodewortfehler} &= 3.57 \cdot 10^{-5} + 1.25 \cdot 10^{-10} + 8.37 \cdot 10^{-17} + 8.99 \cdot 10^{-24} \\
p_{Kodewortfehler} &= 3.57 \cdot 10^{-5}
\end{aligned}$$

2 Informationstheorie und Entropiekodierung

Bei der Entropiekodierung werden die Kodewortlängen in Abhängigkeit der Auftretenswahrscheinlichkeit der Quellenzeichen variiert, wobei Redundanzen im Signal ausgenutzt werden. Die mittlere Kodewortlänge sollte dabei im Idealfall der Entropie entsprechen, so dass die Redundanz minimal ist.

a) Erläutern Sie die der Entropiekodierung zugrundeliegenden Begriffe des Informationsgehaltes, der Entropie, der Mittleren Kodewortlänge und der Redundanz im Sinne der Informationstheorie. Geben Sie jeweils die Berechnungsvorschrift und ein Beispiel aus dem Audiobereich an.
Lösung:
Der Informationsgehalt H_i ist ein Maß für die Unbestimmtheit eines Ereignisses bzw. Zeichens. Je kleiner die Wahrscheinlichkeit eines Zeichens (Audio: Amplitude), desto größer ist H_i.

$$H_i = -\log_2(p_i) \ \ [\text{bit/Zeichen}]$$
$$mit\ p_i \ = \ Auftretenswahrscheinlichkeit\ eines\ Zeichen$$

Die Entropie H_m ist der mittlere Informationsgehalt einer Quelle (Audio: Signal) und gibt die minimale Anzahl Bits je Symbol an, mit der die Quelle eindeutig kodiert werden kann. Sie ist maximal, wenn die Auftretenswahrscheinlichkeiten aller Amplituden gleich sind (z.B. weißes Rauschen) und minimal ($= 0$), wenn es nur eine Amplitude (Gleichspannung) mit der Wahrscheinlichkeit 1 gibt.

$$H_m = \sum_{i=1}^{N} H_i\, p_i \ \ [\text{bit/Zeichen}]$$

Die mittlere Kodewortlänge l_m ergibt sich aus den einzelnen Kodewortlängen, sowie ihren Auftretenswahrscheinlichkeiten:

$$l_m = \sum_{i=1}^{N} l_i\ p_i \ \ [\text{bit/Zeichen}]$$

Im falle einer 16 bit Kodierung eines Audiosignals im Zweier-Komplementär wäre $l_m = 16$ bit/Zeichen, da jedes Kodewort die selbe Länge hat und sich die Auftretenswahrscheinlichkeiten zu 1 summieren. Die Redundanz ist durch die Differenz von mittlerer Kodewortlänge und Entropie gegeben

$$R = l_m - H_m \text{ [bit/Zeichen]} \; .$$

b) Lesen Sie das Audiofile „test.wav" in Matlab ein und requantisieren Sie es auf eine Wortbreite von $w = 3$ bit. Benutzen Sie dafür eine Midtread-Kennlinie.
Lösung:

```
%% b)

y = mean(wavread('test.wav'), 2);

% Quantisierung mit Funktion aus Tutorium 4
w = 3;
[y, ~, ~, codebook] = xquant(y, w, 'mid-tread', 'none');
```

c) Wie ist die Auftretenswahrscheinlichkeiten der 2^w Kodewörter im Quellkode?
Lösung:

```
%% c)

% Zaehle die Haeufigkeit der Amplituden je Quantisierungsstufe
hf = hist(y,codebook);

% Kodewortwahrscheinlichkeiten p_i
p_i = hf/sum(hf)

% Kontrolle: (Summe der p_i muss 1 ergeben)
W_ges = sum(p_i)

hist(y,codebook)
set(gca, 'xTick', codebook, 'xTicklabel', -4:3)
```

Die $2^3 = 8$ Kodeworte und ihre gemessenen Auftretenswahrscheinlichkeiten:

Quellzeichen	x_{-4}	x_{-3}	x_{-2}	x_{-1}	x_0	x_1	x_2	x_3
p_i	0.0001	0.0029	0.0132	0.0934	0.7643	0.1149	0.009	0.0022

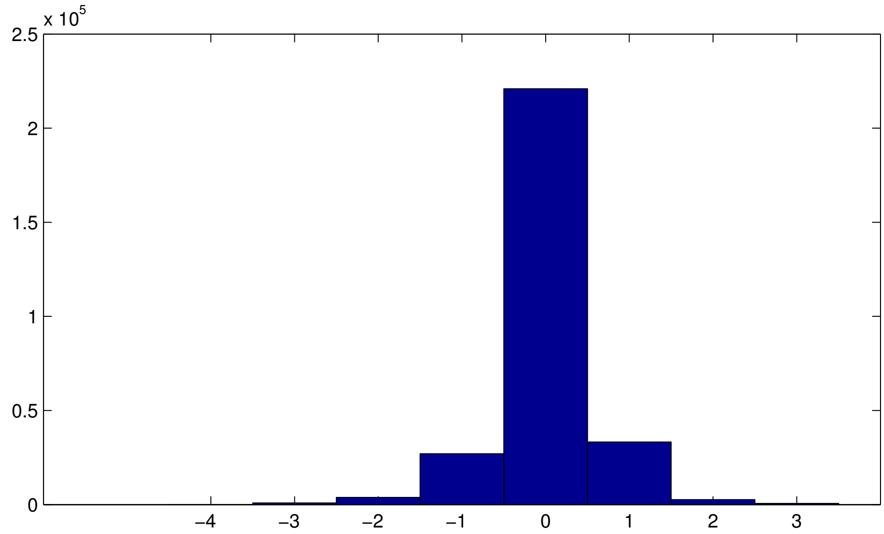


Abbildung 1: Absolute Häufigkeiten der Amplitdudenstufen

d) Wie groß ist die Quellenentropie in bit/Quellenzeichen?

Lösung:

$$H_i = \mathrm{ld}\frac{1}{p_i}$$
$$H_m = \sum_{i=1}^{N} p_i \cdot H_i = 1.11632 \ bit/Zeichen$$

```
%% d)

% Bestimme den Informationsgehalt je Zeichen H_i
H_i = -log2(p_i);

% danach die Entropie H_m:
H_m = sum(p_i.*H_i)
```

e) Konstruieren Sie für diese Quelle Optimalkode nach dem Huffman-Verfahren.

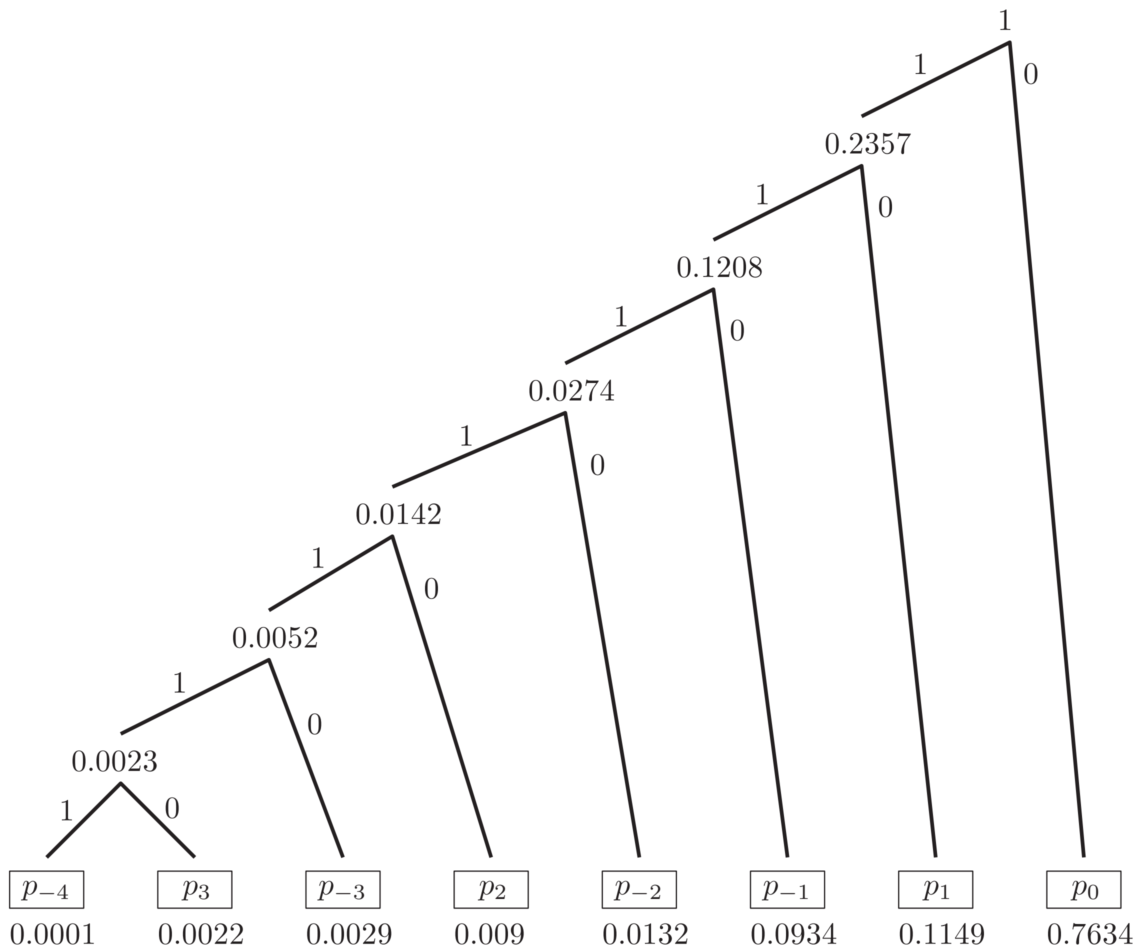
Lösung:

Eigenschaften von Huffman-Kodes:

- Je größer die Auftretenswahrscheinlichkeit eines Kodeworts, desto kürzer ist seine Länge l_i.
- Präfixeigenschaft: Es beginnt kein Kodewort mit der Bitfolge eines anderen, kürzeren Kodewortes.
- Redundanz, nur dann optimal ($= 0$), wenn $p_i = 2^{-n}$

Quellzeichen	x_{-4}	x_{-3}	x_{-2}	x_{-1}	x_0	x_1	x_2	x_3
p_i	0.0001	0.0029	0.0132	0.0934	0.7643	0.1149	0.009	0.0022
Kodewort	1111111	111110	1110	110	1	10	11110	1111110

Hinweis: Ein gut beschriebenes Beispiel für die Erstellung eines Huffmann-Codes finden Sie z.B. hier: `http://www.iti.fh-flensburg.de/lang/algorithmen/code/huffman/huffman.htm`


1
1
0
0.2357
1
0
0.1208
1
0
0.0274
1
0
0.0142
1
0
0.0052
1
0
0.0023
1
0
p_{-4}
0.0001
p_3
0.0022
p_{-3}
0.0029
p_2
0.009
p_{-2}
0.0132
p_{-1}
0.0934
p_1
0.1149
p_0
0.7634

f) Welche mittlere Kodewortlänge ergibt sich aus dem erstellten Kode? Vergleichen Sie die Koderedundanz des Huffman-Kodes mit der eines gleichmäßigen 3-bit-Kodes.
Lösung: Die Mittlere Kodewortlänge berechnet sich aus Kodewortlänge und Auftretenswahrscheinlichkeit der Kodewörter.

$$l_m = \sum_{i=1}^{N} p_i \cdot l_i = 1.4056 \; bit/Zeichen$$

```
%% f)

% mittele Kodewortl?nge des Huffman-Kodes:
l_i = [7 6 4 3 1 2 5 7];

l_m = sum(p_i.*l_i)
```

Als Redundanz bezeichnet man die Differenz zwischen der mittleren Kodewortlänge und der Entropie.

Koderedundanz des Huffman-Kodes

$$R_{k,\,Huffman} = l_m - H_m = 1.4056 - 1.1632 = 0.2424 \; bit/Zeichen$$

Koderedundanz des gleichmäßigen Kodes

$$R_{k,\,3\,bit} = l_m - H_m = 3 - 1.1632 = 1.8368 \; bit/Zeichen$$

g) Wie groß ist der auf diese Weise erzielte „Kompressionsfaktor“, d.h. um welchen Faktor ist die mittlere Kodewortlänge des Huffman-Kodes geringer als die des ursprünglichen Quellkodes?
Lösung:

$$c = \frac{l_{3\,bit}}{l_{Huffman}} = \frac{3}{1.4056} = 2.13$$