Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik

Vorlesung am 9.2.2023

Torsten Ueckerdt | 9. Februar 2023

Thema dieses Kapitels

Informationstheorie hat Anwendungen in

- Quellkodierung
- Kanalkodierung
- Kryptographie

Thema dieses Kapitels

Informationstheorie hat Anwendungen in

- Quellkodierung
  - Reduktion von Redundanz/Irrelevanz am Ausgang einer Informationsquelle
  - Hauptaufgabe: Datenkompression
  - Unterscheidung: Verlustfrei vs. verlustbehaftete Kompression
  - Hohe wirtschaftliche Bedeutung
- Kanalkodierung
- Kryptographie

Thema dieses Kapitels

Informationstheorie hat Anwendungen in

- Quellkodierung
- Kanalkodierung
  - Übertragung von digitalen Daten über gestörte Kanäle
  - Schutz vor Übertragungsfehlern durch Redundanz
  - Fehlerkorrektur
- Kryptographie

Thema dieses Kapitels

Informationstheorie hat Anwendungen in

- Quellkodierung
- Kanalkodierung
- Kryptographie
  - Informationssicherheit:
  - Konzeption, Definition und Konstruktion von Informationssystemen, die widerstandsfähig gegen unbefugtes Lesen und Verändern sind
  - Kryptographie bildet zusammen mit Kryptoanalyse die Kryptologie.

Material für Informationstheorie

- Vorlesungsfolien
- TGI-Skript von Prof. Müller-Quade aus dem WS 08/09
  (auf der TGI-Homepage verlinkt)
- Martin Werner: Information und Kodierung, VIEWEG TEUBNER, 2008

Information

- Sei $\Sigma = \{1, \ldots, n\}$ eine Menge von Zeichen mit Wahrscheinlichkeiten $\{p_1, \ldots, p_n\}$.
- Wir betrachten eine Informationsquelle X, die Zeichen $i \in \Sigma$ mit Wahrscheinlichkeit p_i liefert.

Bemerkung: X wird auch diskrete, endliche Zufallsvariable genannt.

Information

- Sei $\Sigma = \{1, \ldots, n\}$ eine Menge von Zeichen mit Wahrscheinlichkeiten $\{p_1, \ldots, p_n\}$.
- Wir betrachten eine Informationsquelle X, die Zeichen $i \in \Sigma$ mit Wahrscheinlichkeit p_i liefert.

Bemerkung: X wird auch diskrete, endliche Zufallsvariable genannt.

Beispiel

- Ein idealer Würfel wird durch die Wahrscheinlichkeiten $(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$ dargestellt.
- Das Ergebnis des idealen Würfels ist schwer vorherzusagen.
- Der Erkenntnisgewinn nach Ausgang des Experiments ist deshalb groß.

Information

- Sei $\Sigma = \{1, \ldots, n\}$ eine Menge von Zeichen mit Wahrscheinlichkeiten $\{p_1, \ldots, p_n\}$.
- Wir betrachten eine Informationsquelle X, die Zeichen $i \in \Sigma$ mit Wahrscheinlichkeit p_i liefert.

Bemerkung: X wird auch diskrete, endliche Zufallsvariable genannt.

Beispiel

- Betrachte den gezinkten Würfel mit Wahrscheinlichkeiten $(\frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{2})$.
- Hier ist schon klarer, welche Zahl als nächstes gewürfelt wird.
- Der Erkenntnisgewinn ist also kleiner.

Information

- Sei $\Sigma = \{1, \ldots, n\}$ eine Menge von Zeichen mit Wahrscheinlichkeiten $\{p_1, \ldots, p_n\}$.
- Wir betrachten eine Informationsquelle X, die Zeichen $i \in \Sigma$ mit Wahrscheinlichkeit p_i liefert.

Bemerkung: X wird auch diskrete, endliche Zufallsvariable genannt.

Frage

- Wir suchen ein Maß für den Erkenntnisgewinn nach Ausgang k mit Wahrscheinlichkeit p_k.
- Wir bezeichnen diesen Erkenntnisgewinn als **Information** I_{p_k}

Information

- Sei $\Sigma = \{1, \ldots, n\}$ eine Menge von Zeichen mit Wahrscheinlichkeiten $\{p_1, \ldots, p_n\}$.
- Wir betrachten eine Informationsquelle X, die Zeichen $i \in \Sigma$ mit Wahrscheinlichkeit p_i liefert.

Bemerkung: X wird auch diskrete, endliche Zufallsvariable genannt.

Wünsche an die Definition von Information

- Information soll nicht negativ sein. In Formeln: $I_{p_i} \geq 0$
- Ein sicheres Ereignis (also $p_i = 1$) soll keine Information liefern.
- Kleine Änderungen an der Wahrscheinlichkeit sollen nur kleine Änderungen an der Information bewirken.
  Etwas mathematischer ausgedrückt: Information soll stetig sein.

Information

- Sei $\Sigma = \{1, \ldots, n\}$ eine Menge von Zeichen mit Wahrscheinlichkeiten $\{p_1, \ldots, p_n\}$.
- Wir betrachten eine Informationsquelle X, die Zeichen $i \in \Sigma$ mit Wahrscheinlichkeit p_i liefert.

Bemerkung: X wird auch diskrete, endliche Zufallsvariable genannt.

Wünsche an die Definition von Information

- Eine doppelt so lange Zeichenkette soll doppelte Information enthalten
- Deshalb fordern wir, dass $I_{p_i \cdot p_j} = I_{p_i} + I_{p_j}$
- Dies soll später sicherstellen, dass die Information einer (unabhängigen) Zeichenkette gleich der Summe der Einzelinformationen ist.

Information

- Sei $\Sigma = \{1, \ldots, n\}$ eine Menge von Zeichen mit Wahrscheinlichkeiten $\{p_1, \ldots, p_n\}$.
- Wir betrachten eine Informationsquelle X, die Zeichen $i \in \Sigma$ mit Wahrscheinlichkeit p_i liefert.

Bemerkung: X wird auch diskrete, endliche Zufallsvariable genannt.

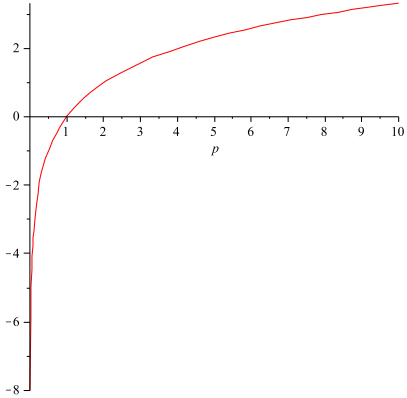
Definition.

Sei p eine Wahrscheinlichkeit. Die Information von p (zur Basis b) ist

$$I_p = \log_b(\frac{1}{p}) = -\log_b(p)$$

Im Folgenden verwenden wir immer die Basis $b = 2$.

Wiederholung: Rechenregeln Logarithmus



- $\log_a(1) = 0$
- $\log_a(x \cdot y) = \log_a(x) + \log_a(y)$
- $\log_a(1/x) = -\log_a(x)$

Basiswechsel:

- $\log_a(x) = \frac{\log_b(x)}{\log_b(a)}$

Information:

- $I_p = \log_2(1/p) = -\log_2(p)$

Information

Definition.

Sei p eine Wahrscheinlichkeit. Die Information von p (zur Basis b) ist

$$I_p = \log_b(\frac{1}{p}) = -\log_b(p)$$

Im Folgenden verwenden wir immer die Basis $b = 2$.

Beispiel 2:

- Betrachte einen Münze mit Seiten 0, 1 und Wkten $p_0 = p_1 = \frac{1}{2}$.
- Die Information eines Münzwurfs ist $\log(1/\frac{1}{2}) = \log(2) = 1$.
- Werfen wir die Münze k mal, so ist die Wahrscheinlichkeit für einen bestimmen Ausgang gleich $\frac{1}{2} \cdot \ldots \cdot \frac{1}{2} = \frac{1}{2^k}$.
- Die Information ist dann $-\log(\frac{1}{2^k}) = \log(2^k) = k$

Entropie

Anschaulich formuliert

- Entropie ist ein Maß für den mittleren Informationsgehalt pro Zeichen einer Quelle.

Interessante andere Sichtweise

- Entropie eines Strings bezeichnet die Länge unter der ein String nicht komprimiert werden kann.
- Die Kolmogorov-Komplexität eines String ist die Länge eines kürzesten Programms, das diesen String ausgibt.
- Damit ist Entropie eine untere Schranke für die Kolmogorov-Komplexität.

Entropie

Definition.

Die Entropie (zur Basis 2) einer diskreten Zufallsvariable X mit Ergebnissen (Zeichen) in Σ und Wahrscheinlichkeiten $p(a) > 0$ für $a \in \Sigma$, ist definiert durch

$$H(X) = \sum_{a \in \Sigma} p(a) \log_2(\frac{1}{p(a)}).$$

Bemerkung:

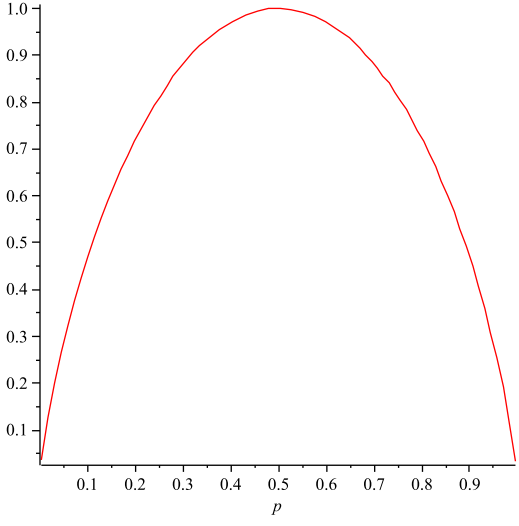
- $H(X) = \sum_{a \in \Sigma} p(a) \cdot I_{p(a)}$
- Also ist die Entropie die erwartete Information einer Auswertung von X.
- Es gilt immer $H(X) \geq 0$.
- Es gilt $H(X) = 0 \Leftrightarrow p(a) = 1$ für ein $a \in \Sigma$.

Bemerkungen zur Entropie

$$H(X) = \sum_{a \in \Sigma} p(a) \log_2(\frac{1}{p(a)})$$

- Die Entropie einer diskreten, endlichen Zufallsvariable mit $|\Sigma| = n$ Zeichen wird maximal, wenn alle Zeichen gleichwahrscheinlich sind: $p(a) = 1/n$ für jedes $a \in \Sigma$.
- Die maximale Entropie beträgt dann
$$H(X) = \sum_{a \in \Sigma} p(a) \log_2(\frac{1}{p(a)}) = n \cdot \frac{1}{n} \cdot \log_2(\frac{1}{1/n}) = \log_2(n).$$
- Die Entropie der deutschen Sprache liegt etwa bei $4,1$.
- Bei 26 Buchstaben ergibt sich eine maximale Entropie von $\log_2(26) \approx 4,7$.

Entropie einer Münze mit Wkt p für Zahl



$$H(X) = \sum_{a \in \Sigma} p(a) \log_2(\frac{1}{p(a)})$$

$$= p \cdot \log_2(\frac{1}{p}) + (1-p) \cdot \log_2(\frac{1}{1-p})$$

(Platzsparende) Kodierungen

- Wir betrachten eine Informationsquelle X, die Zeichen $i \in \Sigma$ mit Wahrscheinlichkeit p_i liefert.
- Zum Kodieren der Zeichen aus Σ haben wir aber nur Zeichenketten aus $\{0,1\}^*$ zur Verfügung.
- Wie können wir Σ ohne Informationsverlust kodieren, dass die erwartete Länge der Ausgabe möglichst klein wird?

Formal

- Wir ordnen jedem Zeichen $i \in \Sigma$ ein Codewort $c_i \in \{0,1\}^*$ zu.
- Wir verwenden keine Trennzeichen.

Beispiel

A: 00 H: 110 L: 10 O: 01

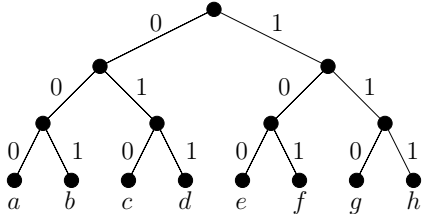
11000101001 ist eine Folge von fünf Codewörtern und steht für:
HALLO

Präfix-Codes

- Bei Codes mit variabler Länge muss man wissen, wann ein neues Codewort beginnt.
- Ein **Präfix-Code** ist ein Code, so dass kein Codewort Anfang eines anderen Codeworts ist.
- Für Präfix-Codes benötigt man deswegen keine Trennzeichen.
- Jeder Präfix-Code kann als Baum dargestellt werden ...

Kodierungsbäume

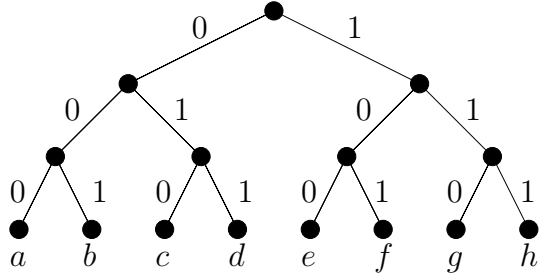
- Wir kodieren im Folgenden binär.
- Sei $\Sigma = \{1, \ldots, n\}$ ein Alphabet mit Präfix-Code $C = \{c_1, \ldots, c_n\}$.
- Der Kodierungsbaum T von (Σ, C) ist ein gerichteter, binärer Baum so dass
  - jede Kante mit 0 oder 1 annotiert ist,
  - ausgehend von einem Knoten höchstens eine Kante mit 0 und höchstens eine Kante mit 1 annotiert ist,
  - die Blätter von T genau die Elemente in Σ sind,
  - der Weg von der Wurzel zu $i \in \Sigma$ mit c_i annotiert ist.



Kodierungsbäume

Beispiele

- Zeichen *b* hat Code 001
- Zeichen *e* hat Code 100
- Zeichen *h* hat Code 111



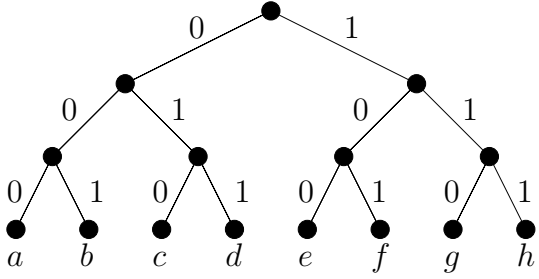
Kodierungsbäume

Beispiele

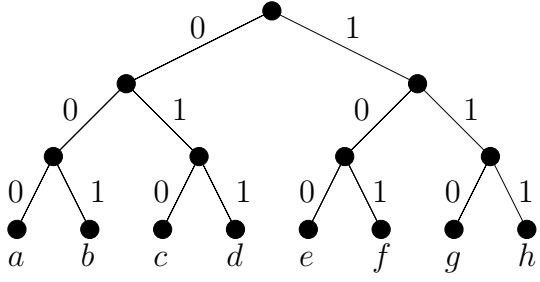
- Zeichen b hat Code 001
- Zeichen e hat Code 100
- Zeichen h hat Code 111

Bemerkungen

- Es besteht ein direkter Zusammenhang zwischen Kodierungen und den zugehörigen Bäumen.
- Die Tiefe $d_T(v)$ eines Knotens v in einem Baum T ist die Anzahl der Kanten auf einem kürzesten Weg von der Wurzel zu v.



Kodierungsbäume

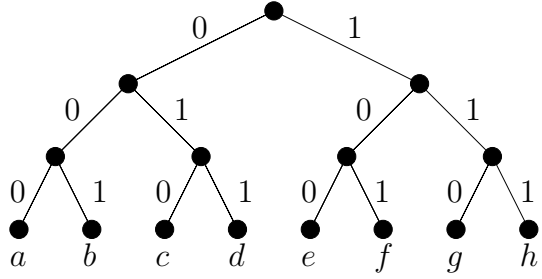


Bemerkungen

- Die Tiefe $d_T(v)$ eines Knotens v in einem Baum T ist die Anzahl der Kanten auf einem kürzesten Weg von der Wurzel zu v.
- Gegeben sei eine Kodierung für Alphabet Σ mit
  - Wahrscheinlichkeit p_i für $i \in \Sigma$,
  - Codewortlänge n_i für $i \in \Sigma$ und
  - zugehörigem Kodierungsbaum T.
- Die mittlere Codewortlänge ist $\overline{n} = \sum_{i \in \Sigma} p_i n_i = \sum_{v \in \Sigma} p_v d_T(v)$.

Beispiel

- Betrachte eine Informationsquelle X mit $\Sigma = \{a, b, c, d, e, f, g, h\}$ und Wahrscheinlichkeiten $p_i = 1/8$ für $i \in \Sigma$.

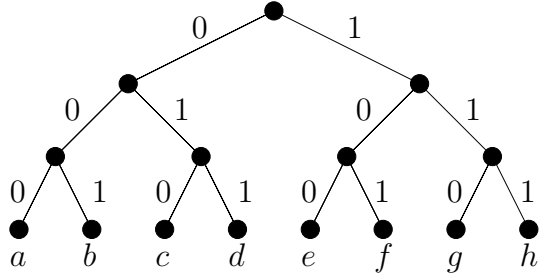


Beispiel

- Betrachte eine Informationsquelle X mit $\Sigma = \{a, b, c, d, e, f, g, h\}$ und Wahrscheinlichkeiten $p_i = 1/8$ für $i \in \Sigma$.

Mittlere Codewortlänge

- Hier haben alle Codewörter Länge 3.
- Die mittlere Codewortlänge ist also 3.



Beispiel

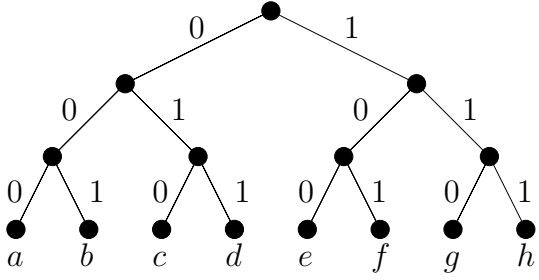
- Betrachte eine Informationsquelle X mit $\Sigma = \{a, b, c, d, e, f, g, h\}$ und Wahrscheinlichkeiten $p_i = 1/8$ für $i \in \Sigma$.

Mittlere Codewortlänge

- Hier haben alle Codewörter Länge 3.
- Die mittlere Codewortlänge ist also 3.

Anzahl der kodierten Zeichen

- Mit jedem zusätzlichen Bit, verdoppelt sich die Größe des darstellbaren Alphabets.
- Um ein Alphabet Σ mit Wörtern gleicher Länge zu kodieren braucht man also $\log_2(|\Sigma|)$ Bits pro Codewort.



Beispiel: Morse-Alphabet

Buchstabe	Morsezeichen	Buchstabe	Morsezeichen
A	○–	N	–○
B	– ○ ○○	O	– – –
C	– ○ –○	P	○ – –○
D	– ○ ○	Q	– – ○–
E	○	R	○ – ○
F	○ ○ –○	S	○ ○ ○
G	– – ○	T	–
H	○ ○ ○○	U	○ ○ –
I	○○	V	○ ○ ○–
J	○ – ––	W	○ – –
K	– ○ –	X	– ○ ○–
L	○ – ○○	Y	– ○ ––
M	––	Z	– – ○○

- Variable Länge
- Kein Präfix-Code
- Zur Unterscheidung von A und ET benötigt man ein Trennzeichen.
- Das Morsealphabet besteht deswegen aus 3 Zeichen.

Quellenkodierungstheorem

- Es seien Codes mit variabler Länge erlaubt.
- Es ist dann nützlich, häufige Zeichen mit kurzen Wörtern zu kodieren.
- Dies verkleinert die mittlere Codewortlänge.

Shannon's Quellenkodierungstheorem.

Sei X eine diskrete endliche Zufallsvariable mit Entropie $H(X)$. Weiter sei ein Präfix-Code für X mit einem Codealphabet aus D Zeichen und minimaler mittlerer Codewortlänge $\overline{n}$ gegeben. Dann gilt

$$\frac{H(X)}{\log_2 D} \leq \overline{n} < \frac{H(X)}{\log_2 D} + 1 \,.$$

Beispiel: Shannon-Fano Kodierung

Zeichen	Wahrscheinlichkeiten	Codewort
0	0,1591	
1	0,1388	
2	0,1125	
3	0,0946	
4	0,0796	
5	0,0669	
6	0,0563	
7	0,0473	
8	0,0398	
9	0,0334	
10	0,0281	
11	0,0237	
12	0,0199	

Beispiel: Shannon-Fano Kodierung

Zeichen	Wahrscheinlichkeiten	Codewort
0	0,1591	0
1	0,1388	0
2	0,1125	0
3	0,0946	0
4	0,0796	1
5	0,0669	1
6	0,0563	1
7	0,0473	1
8	0,0398	1
9	0,0334	1
10	0,0281	1
11	0,0237	1
12	0,0199	1

Beispiel: Shannon-Fano Kodierung

Zeichen	Wahrscheinlichkeiten	Codewort
0	**0,1591**	**00**
1	**0,1388**	**00**
2	**0,1125**	**01**
3	**0,0946**	**01**
4	0,0796	1
5	0,0669	1
6	0,0563	1
7	0,0473	1
8	0,0398	1
9	0,0334	1
10	0,0281	1
11	0,0237	1
12	0,0199	1

Beispiel: Shannon-Fano Kodierung

Zeichen	Wahrscheinlichkeiten	Codewort
0	**0,1591**	**000**
1	**0,1388**	**001**
2	0,1125	01
3	0,0946	01
4	0,0796	1
5	0,0669	1
6	0,0563	1
7	0,0473	1
8	0,0398	1
9	0,0334	1
10	0,0281	1
11	0,0237	1
12	0,0199	1

Beispiel: Shannon-Fano Kodierung

Zeichen	Wahrscheinlichkeiten	Codewort
0	0,1591	000
1	0,1388	001
2	**0,1125**	**010**
3	**0,0946**	**011**
4	0,0796	1
5	0,0669	1
6	0,0563	1
7	0,0473	1
8	0,0398	1
9	0,0334	1
10	0,0281	1
11	0,0237	1
12	0,0199	1

Beispiel: Shannon-Fano Kodierung

Zeichen	Wahrscheinlichkeiten	Codewort
0	0,1591	000
1	0,1388	001
2	0,1125	010
3	0,0946	011
4	**0,0796**	**10**
5	**0,0669**	**10**
6	**0,0563**	**10**
7	**0,0473**	**10**
8	**0,0398**	**11**
9	**0,0334**	**11**
10	**0,0281**	**11**
11	**0,0237**	**11**
12	**0,0199**	**11**

Beispiel: Shannon-Fano Kodierung

Zeichen	Wahrscheinlichkeiten	Codewort
0	0,1591	000
1	0,1388	001
2	0,1125	010
3	0,0946	011
4	**0,0796**	**100**
5	**0,0669**	**100**
6	**0,0563**	**101**
7	**0,0473**	**101**
8	0,0398	11
9	0,0334	11
10	0,0281	11
11	0,0237	11
12	0,0199	11

Beispiel: Shannon-Fano Kodierung

Zeichen	Wahrscheinlichkeiten	Codewort
0	0,1591	000
1	0,1388	001
2	0,1125	010
3	0,0946	011
4	0,0796	1000
5	0,0669	1001
6	0,0563	1010
7	0,0473	1011
8	0,0398	11000
9	0,0334	11001
10	0,0281	11010
11	0,0237	11011
12	0,0199	111000

Ein paar Zwischenschritte später ...

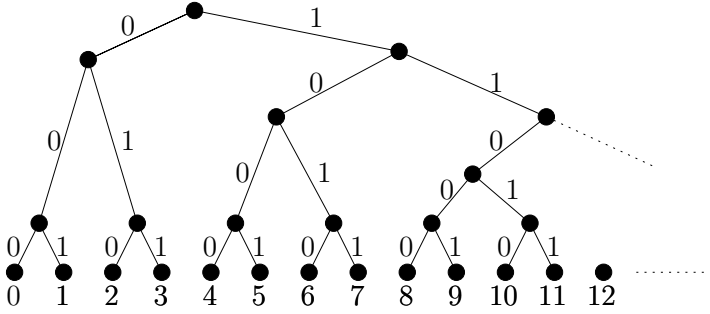
Shannon-Fano Kodierung

Funktion ShannonFano(Z)

- **Eingabe:** Zeichenliste $Z = (z_1, \ldots, z_k)$ mit Wahrscheinlichkeiten $p_1, \ldots, p_k$
- **Ausgabe:** Shannon-Fano Kodierung $(c_1, \ldots, c_k)$
- Wenn $k = 1$
  - return $(c_1 = \epsilon)$ and exit
- Sortiere Zeichen Z absteigend nach Wkt p_i (d.h $p_1 \geq \cdots \geq p_k$).
- Trenne Z in
  - $Z_1 \leftarrow (z_1, \ldots, z_l)$
  - $Z_2 \leftarrow (z_{l+1}, \ldots, z_k)$

  so dass $|\sum_{i=1}^{l} p_i - \sum_{i=l+1}^{k} p_i|$ minimal ist.
- $(c_1, \ldots, c_l) \leftarrow (0s_1, \ldots, 0s_l)$
  mit $(s_1, \ldots, s_l) \leftarrow$ ShannonFano(Z_1)
- $(c_{l+1}, \ldots, c_k) \leftarrow (1s_{l+1}, \ldots, 1s_k)$
  mit $(s_{l+1}, \ldots, s_k) \leftarrow$ ShannonFano(Z_2)
- return $(c_1, \ldots, c_k)$

Kodierungsbaum Shannon-Fano



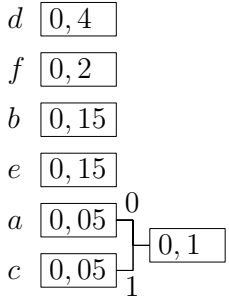
Bemerkung

- Die mittlere Codewortlänge der Shannon-Fano Kodierung muss nicht optimal sein.
- Sie ist deswegen nicht sehr verbreitet.
- Wir werden sehen, dass die **Huffman-Kodierung** optimale mittlere Codewortlänge besitzt.

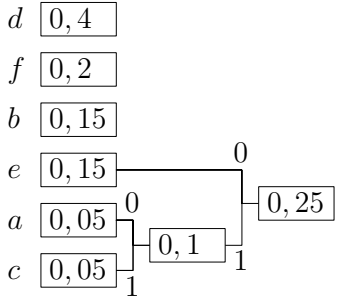
Beispiel: Huffman-Kodierung

d	$0,4$
f	$0,2$
b	$0,15$
e	$0,15$
a	$0,05$
c	$0,05$

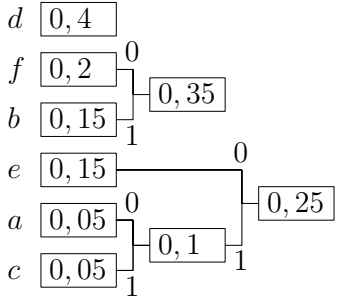
Beispiel: Huffman-Kodierung



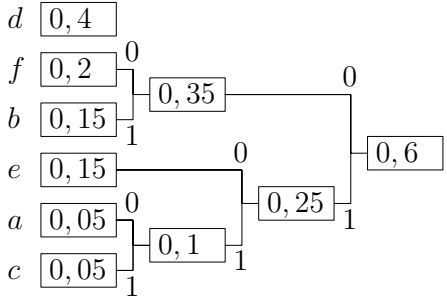
Beispiel: Huffman-Kodierung



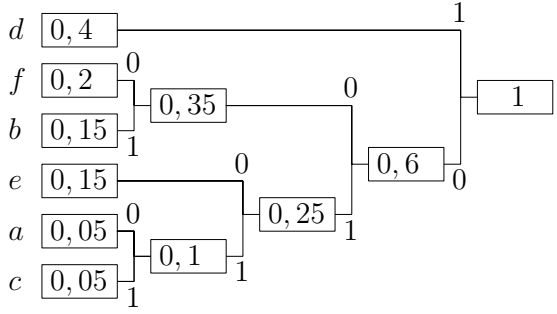
Beispiel: Huffman-Kodierung



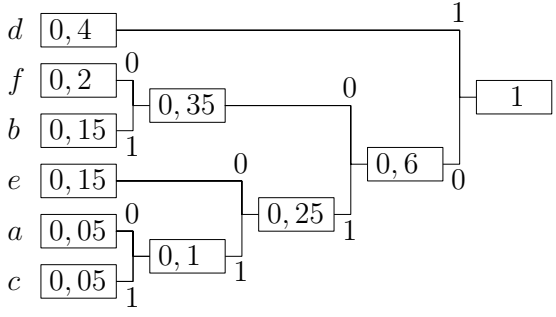
Beispiel: Huffman-Kodierung



Beispiel: Huffman-Kodierung



Beispiel: Huffman-Kodierung



Beispiele

- Zeichen *c* hat Code 0111
- Zeichen *e* hat Code 010
- Zeichen *d* hat Code 1

Huffman-Kodierung

Eingabe: Zeichen $1, \ldots, n$ mit Wahrscheinlichkeiten $p_1, \ldots, p_n$
Ausgabe: Baum T des Huffman-Codes

- Menge $Q = \{1, \ldots, n\}$
- Füge alle Zeichen aus Q als Blätter in T ein
- Für $i = 1, \ldots, n-1$
  - Erzeuge neuen Knoten z für T
  - $u \leftarrow$ extrahiere Element x aus Q mit p_x minimal
  - Bestimme u als linker Nachfolger von z
  - $v \leftarrow$ extrahiere Element x aus Q mit p_x minimal
  - Bestimme v als rechter Nachfolger von z
  - Wahrscheinlichkeit p_z von z ist $p_u + p_v$
  - Füge z in Q ein
- $r \leftarrow$ extrahiere letztes Element aus Q
- return r (r ist Wurzel von T)

Optimalität der Huffman-Kodierung

Satz.

Der Huffman-Algorithmus berechnet einen Kodierungsbaum mit minimaler mittlerer Codewortlänge.

Vorbereitendes Lemma

Lemma.

Sei $\Sigma = \{1, \ldots, n\}$ ein Alphabet mit Wahrscheinlichkeiten P, wobei $P = \{p_1, \ldots, p_n\}$. Seien $x, y \in \Sigma, x \neq y$ eine beliebige Wahl für die zwei unwahrscheinlichsten Zeichen.
Dann gibt es einen Kodierungsbaum T für (Σ, P) mit minimaler mittlerer Codewortlänge, so dass x und y den gleichen Elternknoten besitzen.

Vorbereitendes Lemma: Beweis

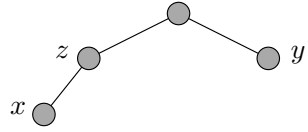
- Sei $\Sigma = \{1, \ldots, n\}$ ein Alphabet mit Wkten $P = \{p_1, \ldots, p_n\}$.
- Seien $x, y \in \Sigma, x \neq y$ eine beliebige Wahl für die zwei unwahrscheinlichsten Zeichen.

Beweis

- Sei T' ein beliebiger Kodierungsbaum für (Σ, P) mit minimaler mittlerer Codewortlänge.
- O.B.d.A gelte für die Tiefe, dass $d'_T(x) \geq d'_T(y)$.
- Sei z der Elternknoten von x in T'.

Fall 1: z hat nur x als Nachkommen

- Dann könnte man z löschen und durch x ersetzen.
- Dieser Baum hätte eine kleinere mittlere Codewortlänge.
- Widerspruch zur Optimalität von T'.



Vorbereitendes Lemma: Beweis

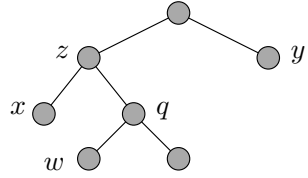
- Sei $\Sigma = \{1, \ldots, n\}$ ein Alphabet mit Wkten $P = \{p_1, \ldots, p_n\}$.
- Seien $x, y \in \Sigma, x \neq y$ eine beliebige Wahl für die zwei unwahrscheinlichsten Zeichen.

Beweis

- Sei T' ein beliebiger Kodierungsbaum für (Σ, P) mit minimaler mittlerer Codewortlänge.
- O.B.d.A gelte für die Tiefe, dass $d'_T(x) \geq d'_T(y)$.
- Sei z der Elternknoten von x in T'.

Fall 2: z hat mehr als 2 Nachkommen

- Sei $w \neq x$ ein Nachfahre von z von maximaler Tiefe.
- Optimalität von T': $p_w \leq p_x$.
- Wahl von x, y: $p_w = p_x$.
- Tausche x mit w. Weiter mit Fall 3.



Vorbereitendes Lemma: Beweis

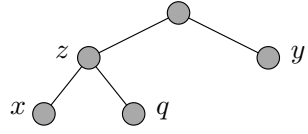
- Sei $\Sigma = \{1, \ldots, n\}$ ein Alphabet mit Wkten $P = \{p_1, \ldots, p_n\}$.
- Seien $x, y \in \Sigma, x \neq y$ eine beliebige Wahl für die zwei unwahrscheinlichsten Zeichen.

Beweis

- Sei T' ein beliebiger Kodierungsbaum für (Σ, P) mit minimaler mittlerer Codewortlänge.
- O.B.d.A gelte für die Tiefe, dass $d'_T(x) \geq d'_T(y)$.
- Sei z der Elternknoten von x in T'.

Fall 3: z hat genau 2 Nachkommen

- Sei $q \neq x$ der andere Nachfahre von z. Tausche q mit y.
- q und y gleiche Tiefe: Tauschen ok.
- q tiefer als y: Wegen Optimalität $p_q = p_y$.



Optimalität der Huffman-Kodierung

Satz.

Der Huffman-Algorithmus berechnet einen Kodierungsbaum mit minimaler mittlerer Codewortlänge.

Beweis

- Wir benutzen Induktion nach der Anzahl der Zeichen $|\Sigma|$ des Alphabets Σ.
- **Induktionsanfang:** Die Aussage ist für ein Zeichen erfüllt.

Beweis – Induktionsschluss

Induktions-Voraussetzung

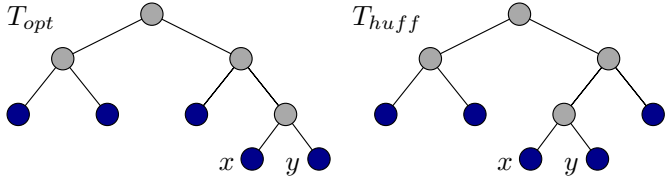
- Der Huffman-Algorithmus berechnet einen optimalen Kodierungsbaum für alle Alphabete Σ mit $|\Sigma| \leq n$ und alle Möglichkeiten für P.

Induktions-Schluss

- Gegeben sei Alphabet $\Sigma = \{1, \ldots, n+1\}$ mit Wahrscheinlichkeiten $P = \{p_1, \ldots, p_{n+1}\}$.
- Für einen Kodierungsbaum T bezeichne $f(T) = \sum_{v \in \Sigma} p_v d_T(v)$ die zugehörige mittlere Wortlänge.
- Wir machen einen Widerspruchsbeweis.
- Bezeichne T_{huff} einen Huffman-Baum für (Σ, P).
- Sei T_{opt} einen Kodierungsbaum für (Σ, P) mit $f(T_{\text{opt}}) < f(T_{\text{huff}})$.

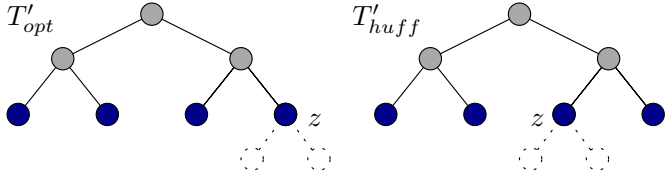
Beweis – Induktionsschluss

- Seien $x, y \in \Sigma$ die Zeichen, die im Huffman-Algorithmus zuerst zusammengefasst werden.
- Vorbereitendes Lemma: Wir können T_{opt} so wählen, dass x und y den gleichen Elternknoten besitzen.



Beweis – Induktionsschluss

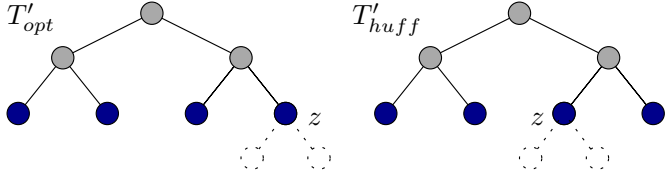
- Seien $x, y \in \Sigma$ die Zeichen, die im Huffman-Algorithmus zuerst zusammengefasst werden.
- Vorbereitendes Lemma: Wir können T_{opt} so wählen, dass x und y den gleichen Elternknoten besitzen.



- Sei $\Sigma' = \Sigma \setminus \{x, y\} \cup \{z\}$ für neues Zeichen z mit Wkt $p_x + p_y$.
- Seien T'_{opt} und T'_{huff} die Bäume, die aus T_{opt} und T_{huff} entstehen, wenn man x, y mit ihrem Elternknoten zu Knoten z verschmilzt.

Beweis – Induktionsschluss

- Sei $\Sigma' = \Sigma \setminus \{x, y\} \cup \{z\}$ für neues Zeichen z mit Wkt $p_x + p_y$.
- Seien T'_{opt} und T'_{huff} die Bäume, die aus T_{opt} und T_{huff} entstehen, wenn man x, y mit ihrem Elternknoten zu Knoten z verschmilzt.

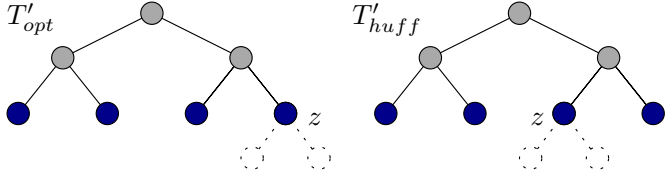


Es gilt:

- T'_{huff} ist ein Huffman-Baum für Instanz Σ' mit den neuen Wkten.
- T'_{opt} ist ein Kodierungs-Baum für Instanz Σ' mit den neuen Wkten.

Beweis – Induktionsschluss

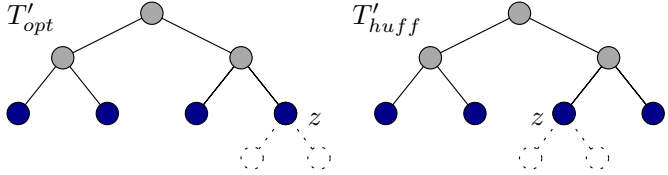
- Sei $\Sigma' = \Sigma \setminus \{x, y\} \cup \{z\}$ für neues Zeichen z mit Wkt $p_x + p_y$.
- Seien T'_{opt} und T'_{huff} die Bäume, die aus T_{opt} und T_{huff} entstehen, wenn man x, y mit ihrem Elternknoten zu Knoten z verschmilzt.



$$
\begin{aligned}
\text{Es gilt: } f(T'_{\text{huff}}) &= f(T_{\text{huff}}) - d_{T_{\text{huff}}}(x)p_x - d_{T_{\text{huff}}}(y)p_y + d_{T'_{\text{huff}}}(z)p_z \\
&= f(T_{\text{huff}}) - p_x - p_y \\
f(T'_{\text{opt}}) &= f(T_{\text{opt}}) - d_{T_{\text{opt}}}(x)p_x - d_{T_{\text{opt}}}(y)p_y + d_{T'_{\text{opt}}}(z)p_z \\
&= f(T_{\text{opt}}) - p_x - p_y
\end{aligned}
$$

Beweis – Induktionsschluss

- Sei $\Sigma' = \Sigma \setminus \{x, y\} \cup \{z\}$ für neues Zeichen z mit Wkt $p_x + p_y$.
- Seien T'_{opt} und T'_{huff} die Bäume, die aus T_{opt} und T_{huff} entstehen, wenn man x, y mit ihrem Elternknoten zu Knoten z verschmilzt.



- Damit ist T'_{opt} ein besserer Coderierungsbaum für Σ' als T'_{huff}.
- Da $|\Sigma'| = n$ ist dies ein Widerspruch zur Induktionsvoraussetzung.

Nachteile der Huffman-Kodierung

- Unterschiedliche Codewortlängen führen zu unterschiedlichen Bitraten und Dekodierungsverzögerung.
- Datenkompression reduziert die Redundanz und erhöht damit die Fehleranfälligkeit.
- Die Kenntnis der Wahrscheinlichkeiten der Zeichen wird vorausgesetzt.
- Universelle Kodierverfahren wie der Lempel-Ziv Algorithmus setzen kein a-priori Wissen an die Statistik der Daten voraus.

Lauflängenkodierung

- Bei der Faxübertragung wird die Vorlage zeilenweise abgetastet und in weiße (w) und schwarze (s) Bildelemente zerlegt.
- Üblicherweise ist die Zahl der weißen Elemente viel höher, als die der schwarzen.
- Wir nehmen der Einfachheit halber an, dass die Bildpunkte voneinander unabhängig sind.
- Bei 15% Schwärzungsgrad ergibt sich eine Entropie von
  $H = -0.85 \cdot \log_2(0.85) - 0.15 \cdot \log_2(0.15) \approx 0.61$
- Bei guter Kodierung sollte eine entsprechende mittlere Codewortlänge zu erwarten sein.

Lauflängenkodierung

- Bei der Faxübertragung wird die Vorlage zeilenweise abgetastet und in weiße (w) und schwarze (s) Bildelemente zerlegt.
- Üblicherweise ist die Zahl der weißen Elemente viel höher, als die der schwarzen.
- Wir nehmen der Einfachheit halber an, dass die Bildpunkte voneinander unabhängig sind.
- Bei 15% Schwärzungsgrad ergibt sich eine Entropie von $H = -0.85 \cdot \log_2(0.85) - 0.15 \cdot \log_2(0.15) \approx 0.61$
- Bei guter Kodierung sollte eine entsprechende mittlere Codewortlänge zu erwarten sein.

Problem:
Wie ist platzsparende Kodierung von einem Alphabet mit zwei Zeichen möglich?

Lauflängenkodierung

Problem:

Wie ist platzsparende Kodierung von einem Alphabet mit zwei Zeichen möglich?

- Möglicher Ansatz: Block-Codes
- Fasse k Zeichen zu Blöcken zusammen und kodiere diesen
- Beispiel $k = 2$:
  - Neues Alphabet: ww,ws,sw,ss.
  - Dieses kann platzsparend kodiert werden.

Beispiel:

Zeichen	ww	ws	sw	ss
Wkt	$\frac{1}{2}$	$\frac{2}{10}$	$\frac{2}{10}$	$\frac{1}{10}$
Huffman	0	11	100	101

Lauflängenkodierung

Lauflängenkodierung

- Spezielle Zusammenfassung für Bildkodierung bei Fax/Videoanwendungen
- Die Länge der Blöcke ist variabel.
- **Idee:** Kodiere nicht die Bildpunkte, sondern den Abstand zwischen zwei schwarzen Bildpunkten.
- Beispiel:

  www *s* ww *s* *s* wwwww *s* w *s* wwwwww *s* wwwwww *s*

  wird aufgefasst als 3204166.
- Für eine Binärkodierung braucht man noch Codes für die Abstände (also für $\mathbb{N}$).
- Um dies platzsparend zu machen, benötigt man Wahrscheinlichkeiten für einzelne Abstände.

Lauflängenkodierung

Lauflängenkodierung

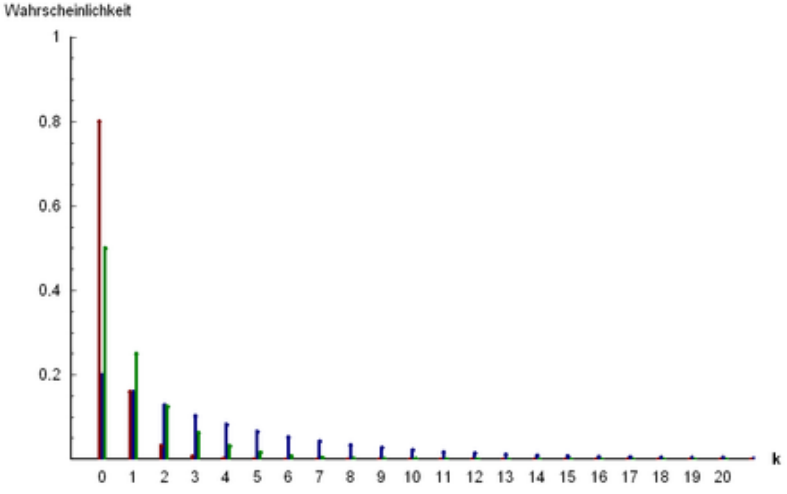
- Wie groß sind die Wkten für die einzelnen Abstände?
- Annahme: Die Bildpunkte sind voneinander unabhängig.
- Sei p_k die Wkt für einen Block aus k aufeinanderfolgenden weißen Bildpunkten mit einem schwarzen Bildpunkt am Schluss

$$p_k = \mathbb{P}(w^k s) = \mathbb{P}^k(w) \cdot \mathbb{P}(s).$$

- Es ergibt sich eine geometrische Verteilung.

Geometrische Verteilung

Quelle: Wikipedia.



Lauflängenkodierung

- Man kann ein Schwarzweißbild über Angabe der Lauflängen verlustfrei rekonstruieren.
- Sonderbehandlung für letzten Block erforderlich.
- Weiteres Problem: Lauflängen können beliebig groß werden.
- Shannon-Fano-Kodierung kann trotzdem einfach angewandt werden.

Abstand	Wkten	Codewort
0	0,1591	000
1	0,1388	001
2	0,1125	010
3	0,0946	011
4	0,0796	1000
5	0,0669	1001
6	0,0563	1010
7	0,0473	1011
8	0,0398	11000
9	0,0334	11001
10	0,0281	11010
11	0,0237	11011
12	0,0199	111000

Kodierung zum Schutz gegen Übertragungsfehler