

Inhaltsverzeichnis

1	Effiziente Algorithmen	2
2	Große Datenbestände indexieren („Information Retrieval“)	3
3	Optimierung	4
4	Informationstheorie	5
4.1	Shannons Informationsmaß	5
4.2	Informationsübertragung und Kanalkapazität	10
4.3	Beweis von Shannons Fundamentalsatz	12
4.4	Optimale Codierung für verlustfreie Kanäle	22
4.5	Anwendungen der Informationstheorie	27

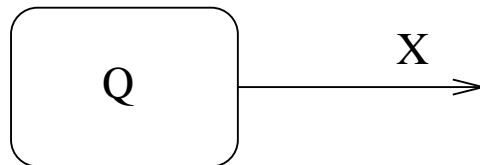
Kapitel 4

Informationstheorie

4.1 Shannons Informationsmaß

Modell:

Eine Informationsquelle $Q = (X, p)$ liefert Zeichen:



Endliches Alphabet $X := \{x_1, x_2, \dots, x_n\}$

Wahrscheinlichkeitsverteilung $p : X \rightarrow [0, 1]$

Es gilt: $\sum_{x_i \in X} p(x_i) = 1$

Kernfrage:

Wie viel „Information“ erhält ein Beobachter von Q ?

Axiome für $H(p(X))$

($H(p)$ = „Unsicherheit von $Q = (x, p)$ “)

1.

$$H\left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}\right) < H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) \Leftrightarrow m < n$$

2.

$$H\left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}\right) + H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) = H\left(\frac{1}{mn}, \frac{1}{mn}, \dots, \frac{1}{mn}\right)$$

3. Für $a = \sum_{i=1}^r p_i$ und $b = \sum_{i=r+1}^n p_i$ (also $a + b = 1$) ist

$$H(p_1, p_2, \dots, p_r, p_{r+1}, \dots, p_n) = H(a, b) + a \cdot H\left(\frac{p_1}{a}, \frac{p_2}{a}, \dots, \frac{p_r}{a}\right) \\ + b \cdot H\left(\frac{p_{r+1}}{b}, \frac{p_{r+2}}{b}, \dots, \frac{p_n}{b}\right)$$

4. $H(p, 1 - p)$ ist stetig in p

(siehe Bild)

Satz:

Die einzige Funktion, die die Axiome 1–4 erfüllt, ist für $c > 0$

$$H(p_1, p_2, \dots, p_n) = -c \sum_{i=1}^n p_i \log p_i$$

(Shannon's Informationsmaß)

Beweis:

siehe Robert Ash: „Information Theory“,
J.Wiley & Sons, 1967 (3.Auflage)

– vgl. Übungen –

Axiom 1:

$Q_1 \rightarrow 489357261\dots$
10 Zeichen gleichverteilt

$Q_2 \rightarrow \text{CABLEFWIFR}\dots$
26 Zeichen gleichverteilt

← hier mehr Unsicherheit,
Kodierungsaufwand etc.

Axiom 2:

$Q_1 \rightarrow 4, 8, 9, 3, 5, 7, 2, \dots$
10 Zeichen gleichverteilt

+

$Q_2 \rightarrow \text{C, A, B, L, F, W, F, } \dots$
26 Zeichen gleichverteilt

enthält genauso
viel Information wie

$Q_3 \rightarrow \text{C4, A8, B9, L3, F5, W7, F2, } \dots$
260 Zeichen gleichverteilt (Tupelbildung)

Axiom 3:

$Q \rightarrow \{x_1, \dots, x_r, x_{r+1}, \dots, x_n\}$

enthält genauso
viel Information wie

$Q_1 \rightarrow \{a, b\}$

+

a: $Q_2 \rightarrow \{x_1, \dots, x_r\}$

b: $Q_3 \rightarrow \{x_{r+1}, \dots, x_n\}$

Abbildung 4.1: Veranschaulichung der Axiome

Bemerkung:

Wahl der Basis des Logarithmus ist beliebig, da über Konstante c noch verfügt werden kann. In der Praxis üblich sind

- (i) $\log_2 = \mathbf{ld} \Rightarrow$ Dimension von H ist Bit/Zeichen
- (ii) $\log_{10} \Rightarrow$ Dimension von H ist Dek/Zeichen

Beispiel:

$$X = \{0, 1\}$$

$$p(0) = p_0$$

$$p(1) = p_1 = 1 - p_0$$

$$\begin{aligned} H(p_0, p_1) &= H(p_0, 1 - p_0) \\ &= -p_0 \cdot \mathbf{ld} p_0 - (1 - p_0) \cdot \mathbf{ld} (1 - p_0) \\ &= f(p_0) \end{aligned}$$

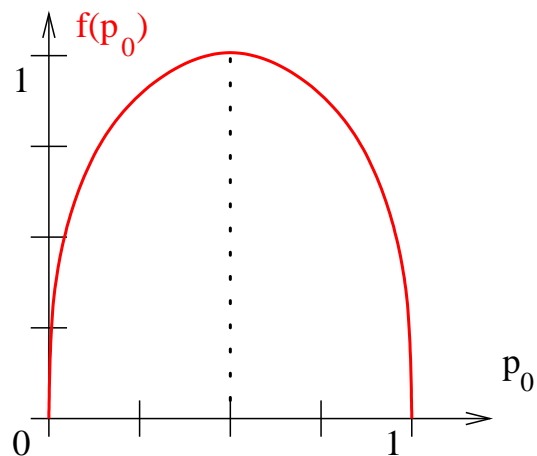


Abbildung 4.2: Informationsmaß einer binären Quelle, abhängig von der Auftretswahrscheinlichkeit eines der beiden Zeichen

Eigenschaften von $H(X)$

1.

$$H(p_1, p_2, \dots, p_n) \leq \log n$$

wobei „=“ genau dann, wenn $p_i = 1/n$ für $i = 1, 2, \dots, n$

2.

$$H(X, Y) = - \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \log p(x_i, y_j)$$

mit $p(x_i, y_j)$ = Wahrscheinlichkeit, dass x_i und gleichzeitig y_j beobachtet wird.

3.

$$H(X, Y) \leq H(X) + H(Y)$$

wobei „=“ genau dann, wenn x_i und y_j unabhängig sind, d.h.

$$p(x_i, y_j) = p(x_i) \cdot p(y_j)$$

allgemein: $H(X_1, X_2, \dots, X_n) \leq H(X_1) + H(X_2) + \dots + H(X_n)$

4. Wie auch $p(y|x)$ = Wahrscheinlichkeit, dass y beobachtet wird, wenn x bereits feststeht („bedingte Wahrscheinlichkeit“):

$$H(Y|X = x_i) := - \sum_{j=1}^m p(y_j|x_i) \log p(y_j|x_i)$$

$$H(Y|X) := \sum_{i=1}^n p(x_i) H(Y|X = x_i)$$

5. Es gilt für alle (!) Quellen X, Y stets:

$$H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$

6.

$$H(Y|X) \leq H(Y)$$

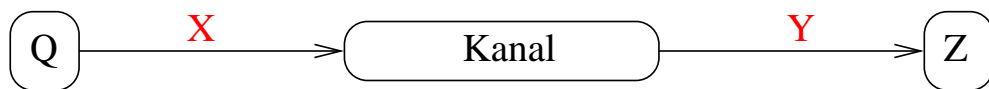
wobei „=“ genau dann, wenn X und Y stochastisch unabhängig sind

4.2 Informationsübertragung und Kanalkapazität

Transinformation I

$$I(X|Y) := H(X) - H(X|Y) = H(Y) - H(Y|X) \quad (\leq H(X))$$

= die durch Kenntnis von Y über X gewonnene Information



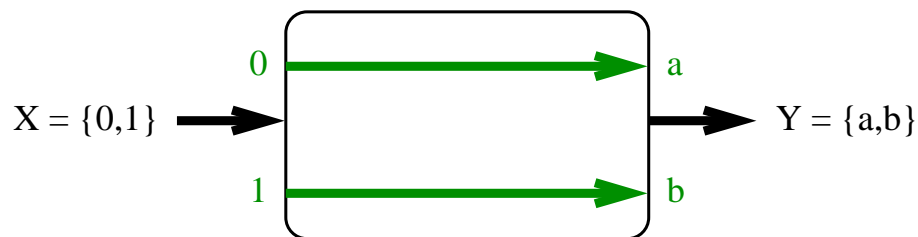
Definition:

Die **Kanalkapazität** C ist definiert durch

$$C := \max_{p(X)} I(X|Y) \quad \text{Einheit: } [C] = \text{Bit/Zeichen}$$

Es wird also die günstige (!) Wahrscheinlichkeitsverteilung $p(X)$ über X angenommen!

Beispiele:



„Verlustfreier Kanal“:

Kenntnis von Y erlaubt vollständige Rekonstruktion von X .

Konsequenz:

$$I(X|Y) = H(X) \Rightarrow H(X|Y) = 0 \quad \text{und damit } C = \text{ld}|X|$$

„Deterministischer Kanal“:

$p(y_j|x_i) \in \{0, 1\}$, d.h. eine Abbildung $f : X \rightarrow Y$ liegt der Informationsübertragung zugrunde.

„Störungsfreier Kanal“:

Kanal ist verlustfrei und deterministisch.

„Nutzloser Kanal“:

$C = 0$

„Binärer symmetrischer Kanal“:

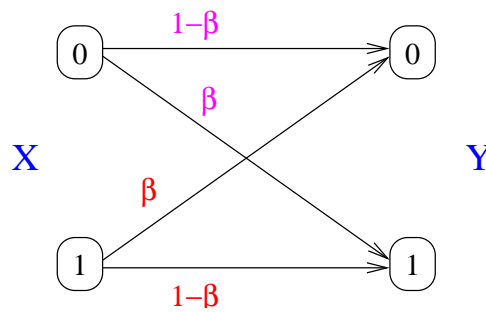
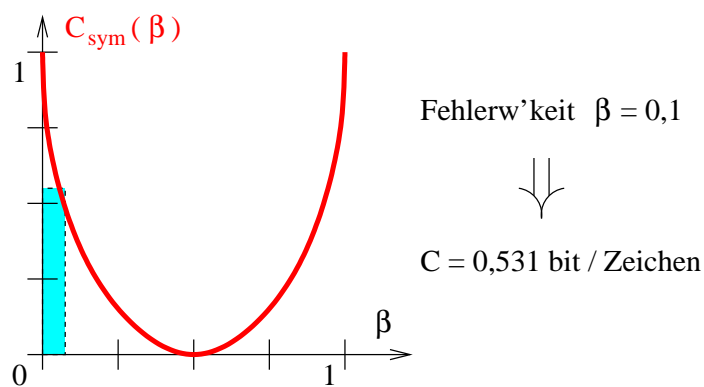


Abbildung 4.3: binärer symmetrischer Kanal mit Fehlerw'keit β

Kapazität:

$$C_{sym} = 1 + \beta \log \beta + (1 - \beta) \log(1 - \beta) \quad \text{mit } 0 \leq \beta \leq 1$$



„Gedächtnisfreier Kanal“:

$$p(y_1 y_2 \dots y_n | x_1 x_2 \dots x_n) = p(y_1 | x_1) \cdot p(y_2 | x_2) \cdot \dots \cdot p(y_n | x_n)$$

d.h. jeder Arbeitstakt ist von vorhergehenden und zukünftigen „statistisch“ unabhängig.
(Endliche Automaten (Mealy, Moore) sind als Kanal deterministisch und *mit* Gedächtnis (Zustände))

Shannons Fundamentalsatz (1948)

Satz:

Sei ein diskreter, gedächtnisfreier Kanal mit Kapazität $C > 0$ und eine positive Zahl R (Rate) mit $R < C$ gegeben. Dann gibt es eine Folge A_1, A_2, \dots von Codes mit folgender Eigenschaft:

Der n -te Code hat $\lfloor 2^{nR} \rfloor$ Codewörter der Länge n und für den maximalen Übertragungsfehler μ_n gilt: $\lim_{n \rightarrow \infty} \mu_n = 0$.

Es ist also möglich, die Fehlerwahrscheinlichkeit beliebig zu reduzieren und dennoch im Mittel R Bit pro Zeichen zu übertragen.

Codierung

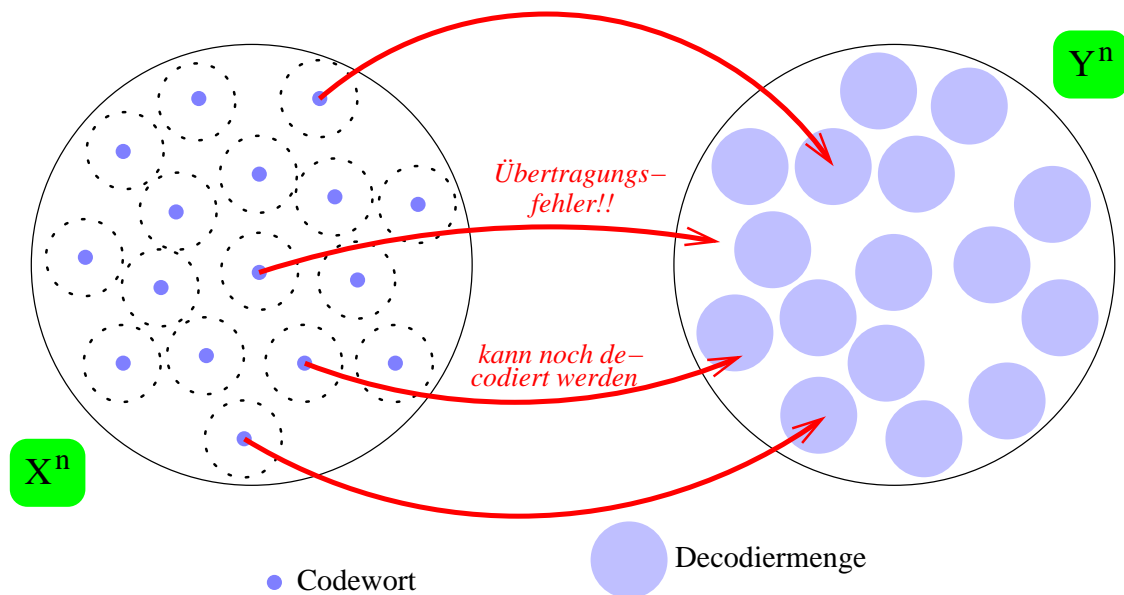
„Random-Coding“: Code-Worte werden per Zufall bestimmt, funktioniert mit Wahrscheinlichkeit 1.

4.3 Beweis von Shannons Fundamentalsatz

Notation:

$$\begin{array}{ll} x \in X^n & y \in Y^n \\ x = x^1 x^2 \dots x^n & y = y^1 y^2 \dots y^n \end{array}$$

$p(y|x)$ = Wahrscheinlichkeit, dass beim „Senden“ von x am Kanalausgang y empfangen wird.



Für diskrete, gedächtnisfreie Kanäle gilt:

$$p(y|x) = p(y^1 y^2 \dots y^n | x^1 x^2 \dots x^n) = \prod_{i=1}^n p(y^i | x^i)$$

Sei eine Quelle $Q(X, q)$ zugrundegelegt, die die Kanalkapazität C erreicht. Dann sei

$q(x)$ = Wahrscheinlichkeit, dass die Quelle $x \in X^n$ ausgibt.

$p'(x, y) :=$ Wahrscheinlichkeit, dass Q x ausgibt und gleichzeitig der Kanal y

$$p'(x, y) = q(x) \cdot p(y|x)$$

Also:

$$\sum_{(x,y) \in X^n \times Y^n} p'(x, y) = 1$$

ist eine Wahrscheinlichkeitsverteilung auf $X^n \times Y^n$.

(Den Strich bei p' lassen wir in Zukunft weg, da zwischen $p(x, y)$ und $p(y|x)$ keine Verwechslungen möglich sind.)

$p(y) := \sum_{x \in X^n} p(x, y) =$ Wahrscheinlichkeit, dass bei Speisung des Kanals durch Q am Ausgang y beobachtet wird.

Definition:

Für festes $R < C$ wird die Menge der typischen Input-Output-Paare $A \subset X^n \times Y^n$ definiert durch

$$A := \left\{ (x, y) \mid \log \frac{p(y|x)}{p(y)} > a := \frac{R+C}{2}n \right\}$$

Solche „typischen“ I/O-Paare werden wir nutzen, um Codes zu konstruieren, wie sie im Fundamentaltheorem genannt werden.

Zum Verständnis der Definition:

Unser binärer symmetrischer Kanal mit Störwahrscheinlichkeit $\beta = 0,1$ hat die Kapazität $C = 0,531$

Wähle $R = 0,469 < C$, also $a = \frac{R+C}{2}n = 0,5 \cdot n$

$X = Y = \{0, 1\}$

Für $x, y \in \{0, 1\}^n$ definieren wir

$$\text{Hamming}(x, y) := \sum_{i=1}^n |x^i - y^i|$$

Alle diejenigen Paare (x, y) sind „typisch“, für die gilt:

$$\text{ld} \frac{p(y|x)}{p(y)} > 0,5 \cdot n$$

Da bei unserem binären symmetrischen Kanal $q(x) = 2^{-n}$ gilt – denn alle x -Sequenzen sind gleichwahrscheinlich – gilt auch $p(y) = 2^{-n}$, denn auch die Ausgabeketten müssen gleichwahrscheinlich sein. Außerdem ist natürlich:

$$p(y|x) = (1 - \beta)^{n-k} \cdot \beta^k = 0,9^{n-k} \cdot 0,1^k$$

wobei $k = \text{Hamming}(x, y)$.

Wie klein muss k (= Zahl der gestörten Bits) sein, damit (x, y) noch „typisch“ ist?

$$\begin{aligned} \text{ld} \frac{p(y|x)}{p(y)} &= \text{ld} (2^n \cdot 0,9^{n-k} \cdot 0,1^k) &> 0,5 \cdot n \\ &= n + (n - k) \cdot \text{ld} 0,9 + k \cdot \text{ld} 0,1 &> 0,5 \cdot n \\ &= n + (n - k) \cdot (-0,152) + k \cdot (-3,322) &> 0,5 \cdot n \end{aligned}$$

Also muß gelten:

$$\begin{aligned}n - 0,152 \cdot n - 0,5 \cdot n &> k \cdot (-0,152 + 3,322) \\0,348 \cdot n &> 3,170 \cdot k \\0,1097 \cdot n &> k\end{aligned}$$

Also: Nur wenig mehr als jedes 10-te Bit darf gestört sein! Wenn (x, y) also „typisch“ sind, so gilt

$$\text{Hamming}(x, y) < 0,1097 \cdot n$$

Weitere Konsequenz: Für festes n (z.B. $n = 10$) kann die Menge A wie folgt konstruiert werden:

$$A = \{(x, y) \mid x \in \{0, 1\}^n \wedge \text{Hamming}(x, y) < 0,1097 \cdot n\}$$

Übungsaufgabe: Berechnen Sie $|A|$ für $n = 15$.

Lemma:

Für jede natürliche Zahl s existiert ein Code (s, n, λ) [wobei $s =$ Anzahl der Codewörter, $\lambda =$ maximale Wahrscheinlichkeit, dass ein gesendetes Codewort falsch dekodiert wird], so dass für den Fehler gilt

$$\lambda \leq s \cdot 2^{-a} + P((x, y) \notin A)$$

wobei $A =$ „typische Paare“ wie oben definiert wurde.

(Dabei ist für X ein $q(X)$ zugrunde zu legen, dass die Kanalkapazität C erreicht. Setze zukünftig $q(x) =: p(x)$)

Anmerkungen:

$$(x, y) \in X^n \times Y^n$$

$$P((x, y) \notin A) + P((x, y) \in A) = 1$$

Wenn wir dieses Lemma bewiesen haben, sind wir fast fertig, denn wir können $s := \lfloor 2^{nR} \rfloor$ setzen und müssen dann nur noch beweisen, dass gilt

$$\lim_{n \rightarrow \infty} \lambda_n = 0$$

Beweis des Lemmas:

Mit Hilfe der Menge $A =$ „typische Paare“ wird ein Code konstruiert. Notation: $\varepsilon := s \cdot 2^{-a} + P((x, y) \notin A)$

Falls $\varepsilon \geq 1$ sein sollte (z.B. für zu große s), so ist nichts mehr zu beweisen. Es interessiert also nur noch $0 < \varepsilon < 1$.

Definition: $A_x := \{y \mid (x, y) \in A\}$

[A_x ist die Menge der y , in die die Input-Sequenz x „typischerweise“ abgebildet wird vom Kanal.]

Den gewünschten Code bauen wir wie folgt auf:

1. Suche (wenn möglich) ein $x_{(1)}$, so dass $P\{y \mid y \in Ax_{(1)}\} \geq 1 - \varepsilon$
 [Wenn ε klein ist, müssen also fast alle durch $x_{(1)}$ =Input am Kanalausgang erzeugten y in $A_{x_{(1)}}$ liegen.]

1. Codewort: $x_{(1)}$

Dekodiermenge: $B_1 := A_{x_{(1)}}$

Dekodierfehler-Wahrscheinlichkeit beim Senden von $x_{(1)}$: $\leq \varepsilon$

2. Suche (wenn möglich) ein $x_{(2)}$, so dass $P\{y \mid y \in Ax_{(2)} \setminus B_1\} \geq 1 - \varepsilon$

2. Codewort: $x_{(2)}$

Dekodiermenge: $B_2 := Ax_{(2)} \setminus B_1$

Dekodierfehler-Wahrscheinlichkeit beim Senden von $x_{(2)}$: $\leq \varepsilon$

und so weiter; es folgt allgemein:

- i . Suche (wenn möglich) ein $x_{(i)}$, so dass $P\{y \mid y \in Ax_{(i)} \setminus B_1 \setminus B_2 \setminus \dots \setminus B_{i-1}\} \geq 1 - \varepsilon$

i -tes Codewort: $x_{(i)}$

Dekodiermenge: $B_i := Ax_{(i)} \setminus \bigcup_{j=1}^{i-1} B_j$

Dekodierfehler-Wahrscheinlichkeit beim Senden von $x_{(i)}$: $\leq \varepsilon$

Ergebnis:

Wegen $B_i \neq \emptyset$ für $i = 1, 2, \dots$ bricht das Verfahren bei einem Index t ($= i_{\max}$) ab.

Codewörter: $x_{(1)}, x_{(2)}, \dots, x_{(t)}$

Dekodiermengen: B_1, B_2, \dots, B_t (disjunkt!)

Gesamt-Dekodiermenge: $B := B_1 \cup B_2 \cup \dots \cup B_t$

Maximale Wahrscheinlichkeit für Dekodierfehler: $\lambda \leq \varepsilon$

[Falls $t = 0$, so setze $B = \emptyset$.]

Der so konstruierte Code soll das Lemma befriedigen. Eigentlich ist schon fast alles erfüllt, allerdings fehlt noch der **Nachweis**, dass $t \geq s$.

Prüfe also

$$\begin{aligned}
 P((x, y) \in A) &= \sum_{(x, y) \in A} p(x, y) = \sum_x p(x) \sum_{y \in A_x} p(y|x) \\
 &= \sum_x p(x) \left[\sum_{y \in B \cap A_x} p(y|x) + \sum_{y \in \overline{B} \cap A_x} p(y|x) \right] \\
 &= \underbrace{\sum_x p(x) \sum_{y \in B \cap A_x} p(y|x)}_{T1} + \underbrace{\sum_x p(x) \sum_{y \in \overline{B} \cap A_x} p(y|x)}_{T2}
 \end{aligned}$$

Abschätzung von T1:

Wegen $B \cap A_x \subset B$ gilt

$$T1 = \sum_x p(x) \sum_{y \in B \cap A_x} p(y|x) \leq \sum_x p(x) \sum_{y \in B} p(y|x) = P(y \in B)$$

Wegen $B_i \subset A_{x_{(i)}}$ gilt

$$P(y \in B) = \sum_{i=1}^t P(y \in B_i) \leq \sum_{i=1}^t P(y \in A_{x_{(i)}})$$

Wenn aber $y \in A_{x_{(i)}}$, dann ist $(x_{(i)}, y) \in A$. Daher gilt

$$\log \frac{p(y|x_{(i)})}{p(y)} > a$$

Nach Delogarithmieren:

$$\begin{aligned}
 \frac{p(y|x_{(i)})}{p(y)} &> 2^a \\
 p(y) &< p(y|x_{(i)}) \cdot 2^{-a}
 \end{aligned}$$

Also ist

$$P(y \in A_{x(i)}) = \sum_{y \in A_{x(i)}} p(y) \leq 2^{-a} \underbrace{\sum_{y \in A_{x(i)}} p(y|x(i))}_{\leq 1} \leq 2^{-a}$$

Es gibt t Codewörter, also ergibt sich aus $P(y \in B) = \sum_{i=1}^t P(y \in B_i) \leq \sum_{i=1}^t P(y \in A_{x(i)})$:

$$T1 \leq P(y \in B) \leq \underbrace{\sum_{i=1}^t P(y \in A_{x(i)})}_{\leq 2^{-a}} \leq t \cdot 2^{-a}$$

Also ist $T1 \leq t \cdot 2^{-a}$

Abschätzung von T2:

$$T2 = \sum_x p(x) \sum_{y \in \overline{B} \cap A_x} p(y|x)$$

Es gilt $\overline{B} \cap A_x = \emptyset$ für Codewörter $x(i)$, wegen der Entstehung der B_i .
Wir behaupten nun

$$\sum_{y \in \overline{B} \cap A_x} p(y|x) < 1 - \varepsilon \quad \text{für jedes } x$$

Für Codewörter $x = x_i$ gilt $\overline{B} \cap A_x = \emptyset$, also ist die Behauptung okay.

Sei nun x kein Codewort. Wenn im g.z. Beh.

$$P\{y \in \overline{B} \cap A_x\} \geq 1 - \varepsilon,$$

so könnte der Code erweitert werden, weil $\overline{B} \cap A_x = A_x \setminus B_1 \setminus B_2 \setminus \dots \setminus B_t$ (Prüfen die Konstruktion des Codes).

Das ist nicht möglich, also

$$T2 = \sum_x p(x) \sum_{y \in \overline{B} \cap A_x} p(y|x) < \sum_x p(x) \cdot (1 - \varepsilon) = (1 - \varepsilon)$$

Zusammengefaßt gilt also

$$P((x, y) \in A) \leq t \cdot 2^{-a} + (1 - \varepsilon)$$

Also

$$\varepsilon \leq t \cdot 2^{-a} + P((x, y) \notin A) \quad \Rightarrow \quad t \geq s$$

Wie haben also einen Code mit mindestens s Codeworten und Fehlerwahrscheinlichkeit

$$\lambda \leq \varepsilon = s \cdot 2^{-a} + P((x, y) \notin A)$$

Damit ist das Lemma bewiesen.

Beweis des Fundamentalsatzes mit Hilfe des Lemmas:

Wähle $s = \lfloor 2^{nR} \rfloor$, $a = n(R + C)/2 < n \cdot C$.

Aus dem Lemma folgt, dass es einen Code $(\lfloor 2^{nR} \rfloor, n, \lambda_n)$ gibt mit

$$\lambda_n \leq \lfloor 2^{nR} \rfloor \cdot 2^{-n(R+C)/2} + P((x, y) \notin A)$$

Der erste Term konvergiert für $n \rightarrow \infty$ gegen Null (wegen $R - C < 0$):

$$\lim_{n \rightarrow \infty} 2^{n(R-(R+C)/2)} = \lim_{n \rightarrow \infty} 2^{n(R-C)/2} \rightarrow 0$$

Bleibt noch zu zeigen, dass auch

$$\lim_{n \rightarrow \infty} P((x, y) \notin A) = 0$$

Es gilt

$$p((x, y) \notin A) = P\left\{(x, y) \mid \log \frac{p(y|x)}{p(y)} \leq a\right\} \quad (\text{siehe Def. von A})$$

Wegen $p(y|x) = \prod_{i=1}^n w(y^i|x^i)$ [hier ist w die Übertragungswahrscheinlichkeit des Kanals] und $p(y) = \prod_{i=1}^n w(y^i)$ gilt

$$\log \frac{p(y|x)}{p(y)} = \sum_{i=1}^n \underbrace{\log \frac{w(y^i|x^i)}{w(y^i)}}_{\text{Zufallsvariable } V(x^i, y^i)}$$

Zufallsvariable $V(x^i, y^i)$ funktional abgeleitet aus den Zeichen auf der i -ten Position ...

Die $V(x^i, y^i)$ sind unabhängig voneinander und identisch verteilt. Der Erwartungswert von $V(x^i, y^i)$ ist:

$$\begin{aligned}
 \sum_{x_i, y_i} w(x_i, y_i) \log \frac{w(y^i|x^i)}{w(y^i)} &= H(Y_i) - H(Y_i|X_i) = C, \quad \text{weil ...} \\
 &= \sum_{x_i, y_i} w(x_i, y_i) (\log w(y^i|x^i) - \log w(y^i)) \\
 &= \underbrace{\sum_{x_i, y_i} w(x_i, y_i) \log w(y^i|x^i)}_{T_1} - \underbrace{\sum_{x_i, y_i} w(x_i, y_i) \log w(y^i)}_{T_2} \\
 T_1 &= \sum_{x_i, y_i} w(x_i) \cdot w(y_i|x_i) \log w(y^i|x^i) \\
 &= - \sum_{x_i, y_i} w(x_i) \cdot H(Y|X = x_i) \\
 &= -H(Y|X) \\
 T_2 &= - \sum_{x_i, y_i} w(x_i, y_i) \log w(y_i) \\
 &= - \sum_{x_i, y_i} w(x_i|y_i) \cdot w(y_i) \log w(y_i) \\
 &= - \sum_{y_i} w(y_i) \log w(y_i) \\
 &= H(Y)
 \end{aligned}$$

Das „Gesetz der großen Zahl“ besagt nun, dass das arithmetische Mittel von n unabhängigen, gleich verteilten Zufallsvariablen gegen den Erwartungswert konvergiert. Also:

$$P((X, Y) \notin A) = P\left(\frac{1}{n} \sum_{i=1}^n V(x_i, y_i) \leq \frac{R+C}{2}\right)$$

Da $(R+C)/2 < C$ ist, konvergiert $P((x, y) \notin A)$ für $n \rightarrow \infty$ gegen Null.

Damit ist also $\lim_{n \rightarrow \infty} \lambda_n \rightarrow 0$ bewiesen und **wir haben Shannons Fundamentalsatz bewiesen!**

Anmerkungen

Wenn man Lemma und Fundamentalsatz auf den binären Kanal mit $\beta = 0, 1$ (Störungswahrscheinlichkeit) anwendet, ergibt sich:

- Die Mengen $A_x := \{y \mid (x, y) \in A\}$ sind $A_x := \{y \mid \text{Hamming}(x, y) \leq \lfloor 0,1097 \cdot n \rfloor\}$.
Im Mittel sind aber bei der Übertragung von x durch den Kanal nur $\lfloor 0,1 \cdot n \rfloor$ Bit gestört. Für $n \rightarrow \infty$ geht (wegen des Gesetzes der großen Zahlen) also die Wahrscheinlichkeit dafür, dass für das empfangene y gilt $y \notin A_x$ gegen Null, auch $\lim_{n \rightarrow \infty} P((x, y) \notin A) = 0$.

- An sich kann man mit folgendem Verfahren Codes konstruieren:

1. Suche im $E^n = \{0, 1\}^n$, $r < 2^{nR}$ Sende-Codewörter x_1, x_2, \dots, x_r so aus, dass $\forall i, j : \text{Hamming}(x_i, x_j) \geq 2 \cdot e + 1$ und e maximal. (\rightarrow Optimierungsproblem!)
2. Definiere die Dekodiermengen B_i ($i = 1, \dots, r$) wie folgt:
 $B_i := \{y \mid \text{Hamming}(x_i, y) \leq e\}$
(Also: $B_i \cap B_j = \emptyset$ für $i \neq j$)
3. Die Wahrscheinlichkeit, dass bei der Übertragung eines x_i durch den Kanal bis zu e Fehler-Bits beobachtet werden, ist:

$$w(0\dots e \mid x_i) = \sum_{i=0}^e \binom{n}{i} (1 - \beta)^{n-i} \cdot \beta^i$$

Die Fehlerwahrscheinlichkeit ist also $\lambda_n = 1 - w(0\dots e \mid x_i)$

Mit Wahrscheinlichkeit $w_e = \binom{n}{e+1} (1 - \beta)^{n-e-1} \cdot \beta^{e+1}$ wird ein Fehler erkannt, der nicht korrigiert werden kann; empfangenes y liegt zwischen zwei Dekodiermengen.

Wie viel „Information“ überträgt ein Code $(n, 2^{nR}, \lambda)$ im Grenzfall $\lambda = 0$?

Im Prinzip sind die Codeworte „Super-Alphabetzeichen“:

Bei 2^{nR} gleichwahrscheinlichen „Zeichen“ hat der „Super-Kanal“ eine Kapazität von $\text{ld}(2^{nR}) = nR$, also pro Einzelzeichen $n \cdot R/n = R = \text{Rate}$.

Fehlererkennung und Redundanz

Redundanz = überflüssige Information bei einer Codierung.

Binärer symmetrischer Kanal : Informationsgehalt eines Codeworts nominal n bit
Nutzinformation : $n \cdot R$ bit
Redundanz : $n(1 - R)$ bit

Satz: („strong converse“ zum Fundamentaltheorem)

Für jede Folge $A_n = (\lfloor 2^{nR} \rfloor, n, \lambda_n)$ von Codes mit $R > C$ gilt:
 $\lim_{n \rightarrow \infty} \lambda_n = 1$.

Beweis:

Hier nur die Idee, wie es z.B. beim binären symmetrischen Kanal gemacht wird:

Wenn die Dekodiermengen den „Radius“ r haben, so gilt

$$\sum_{j=0}^r \binom{n}{j} \beta^j \cdot (1 - \beta)^{n-j} \geq 1 - \lambda$$

Damit sind für jedes Codewort im E^n

$$\sum_{j=0}^r \binom{n}{j} = |B_i| \quad \text{Elemente blockiert.}$$

Der E^n hat aber nur 2^n Elemente, also muss gelten

$$\frac{2^n}{\sum_{j=0}^r \binom{n}{j}} \geq \lfloor 2^{nR} \rfloor$$

und das geht nicht auf!

4.4 Optimale Codierung für verlustfreie Kanäle

Gegeben:

z.B. ein verlustfreier binärer Kanal, $0 \rightarrow 0$ und $1 \rightarrow 1$ (deterministisch), n Einzelnachrichten.

x_1, x_2, \dots, x_n , die mit den Wahrscheinlichkeiten $p(x_i)$, $\sum_i p(x_i) = 1$, auftreten und zu übertragen sind.

Wie muss codiert werden, um möglichst viel Information durch den Kanal zu bringen?

Lösung:

Codewörter *variabler* Länge, also zumeist keine Block-Codes,

$x_1 \rightarrow 010$
z.B. $x_2 \rightarrow 110$
 $x_3 \rightarrow 10$

Eindeutig entzifferbarer Code = Jede Sequenz von Codewörtern kann
(*Minimalanforderung*) zurückkonvertiert werden. (Existenz-Test)

Sofort entzifferbarer Code = Jede Sequenz von Codewörtern kann
von links nach rechts – ohne Vorausblick –
entschlüsselt werden.

Kriterium: Kein Codewort ist Präfix eines anderen,
z.B. $\underbrace{10}_{\text{10}} \underbrace{110}_{\text{110}} \underbrace{010}_{\text{010}} \underbrace{010}_{\text{010}} \underbrace{10}_{\text{10}} \underbrace{10}_{\text{10}} \underbrace{110}_{\text{110}}$

Es gibt eindeutig entzifferbare, aber nicht sofort entzifferbare
Codes, z.B. $x_1 \rightarrow 010$ $x_2 \rightarrow 011$ $x_3 \rightarrow 01$
z.B. die Folge 011010101001001101

Satz:

Sei eine Quelle $Q = (X, p)$ mit $H(X)$ gegeben. Sei ferner ein
verlustfreier ungestörter Kanal mit D Kanalsymbolen gegeben.
Dann existiert ein sofort entzifferbarer Code für X , für dessen
mittlere Codewortlänge

$$\bar{n} := \sum_{i=1}^n p(x_i) \cdot L_i \quad (L_i \text{ ist Codewortlänge für } x_i)$$

gilt:

$$\frac{H(X)}{\log D} \leq \bar{n} < \frac{H(X)}{\log D} + 1 \quad \text{meist } D = 2, \Rightarrow \log D = 1$$

(beim binären Kanal: $H(X) \leq \bar{n} < H(X) + 1$)

Praktische Konsequenz:

$H(X)$ ist auch ein Maß für den Codierungsaufwand. (ungestörte Codierung)

Definition:

Ein eindeutig und sofort entzifferbarer Code $f : X \rightarrow \{0, 1\}^*$ heißt **optimal**, wenn die mittlere Codewortlänge

$$\bar{n} := \sum_{x \in X} p(x) \cdot l(f(x))$$

minimal ist. (Hier: $l(\dots) :=$ Länge einer Zeichenkette)

Aussage:

Eine Sequenz von k Zeichen aus einer Quelle $Q = (X, p)$ kann im Mittel mit $k \cdot \bar{n}$ Bit codiert werden.

Problem: Gegeben: X, p .

Wie kann mit einem systematischen Verfahren ein optimaler, sofort entzifferbarer Code konstruiert werden?

Satz von Huffman:

Sei w_1, w_2, \dots, w_n aus $\{0, 1\}^*$ ein optimaler, sofort entzifferbarer Code für $1 > p_1 \geq p_2 \geq \dots \geq p_n > 0$.

Sei

$$p_i'' + p_i' = p_i$$

eine Zerlegung von p_i , so dass $p_n \geq p_i' \geq p_i'' > 0$. Dann ist

$$w_1, w_2, \dots, w_{i-1}, w_{i+1}, \dots, w_n, w_i 1, w_i 0$$

ein optimaler, sofort entzifferbarer Code für $p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_n, p_i', p_i''$.

Anwendungen:

- Speicherung großer Informationsmengen mit geringstmöglichen Speicheraufwand, ungestört.
- Codierung von Befehlen in Rechnern
- Codierung von Kommandos mit dem Ziel, Tippaufwand zu sparen.

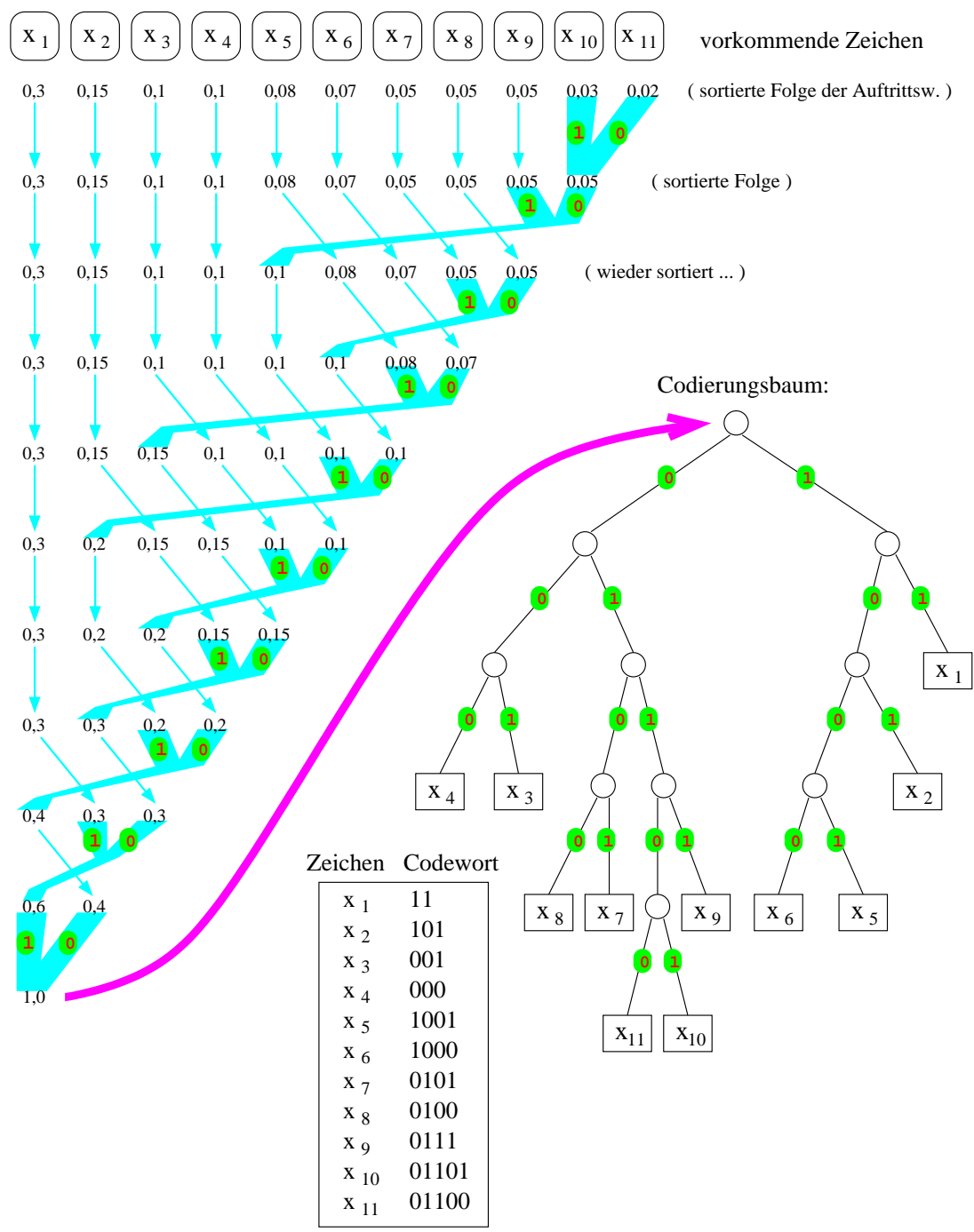


Abbildung 4.4: Durchführung des Verfahrens von Huffman

Beweis:

Annahme, der neue Code

$$\begin{aligned} w_1, w_2, \dots, w_{i-1}, w_{i+1}, \dots, w_n, w_i 1, w_i 0 \\ p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_n, p'_i, p_i \quad (p_i + p'_i = p_i) \end{aligned}$$

wäre *nicht* optimal.

Suche einen optimalen Code, z.B.

$$C' : w'_1, w'_2, \dots, w'_{i-1}, w'_{i+1}, \dots, w'_n, w'_{n+1}, w'_{n+2}$$

Es gilt für optimale, sofort entzifferbare Codes:

1. $p_i > p_k \rightarrow |w_i| \leq |w_k| = \text{Länge von } w_k$
2. $|w_{n+1}| = |w_{n+2}|$, sonst könnte 1 Bit weggelassen werden.
3. Es gibt ein Codewort w_j mit $|w_j| = |w_{n+2}|$
und $w_j = \overline{w}e_j, w_{n+2} = \overline{w}e_2$,
d.h. w_j und w_{n+2} haben gleich langen Präfix.
(Wenn nämlich keine 2 Präfixe übereinstimmen, könnten alle letzten Bits gestrichen werden, da Unterscheidung schon vorher möglich!)

\Rightarrow Vertausche w_{n+1} mit einem gleichlangen Codewort, so dass w_{n+1} und w_{n+2} gleichen Präfix, also: $w'_{n+1} = \overline{w}0, w_{n+2} = \overline{w}1$
Verschmelze p'_i und p_i und konstruiere rückwärts:
 $\overline{C}' : w'_1, w'_2, \dots, \overline{w}_i, \dots, w'_n$
Der Code \overline{C}' muss besser sein als w_1, w_2, \dots, w_n .

\Rightarrow Widerspruch zur Voraussetzung.

Für den Code aus der Abbildung ist die durchschnittliche Codewortlänge:

$$\bar{n} := \sum_{i=1}^{11} p_i \cdot |w_i| = 3,10 \text{ bit/Codewort}$$

$$H := \sum_{i=1}^{11} -p_i \cdot \text{ld}(p_i) = 3,0690149 \text{ bit/Codewort}$$

4.5 Anwendungen der Informationstheorie

Telefax

„Run Length Coding“

i_1, i_2, i_3, \dots optimal zu codieren, wobei $i_j \in \{1, 2, 3, \dots\}$

Methode: $i = 8$ entspricht binär (2-adisch) 1000_2

Run Length: $1 \underline{0} \underline{0} \underline{0} \underline{0} \underline{0} \underline{0} \underline{1}$
(Die unterstrichenen Bits sind die Füllbits)

Ergebnis: $n(i) = O(\text{ld}(i)) =$ „Länge des Codewortes für i “
(kann noch erheblich verbessert werden)

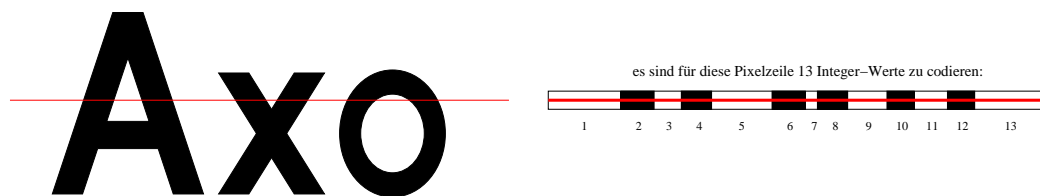


Abbildung 4.5: Verfahren beim Telefax: Die Pixelzeilen werden *run-length*-codiert

Telefax-Codes sind (fast) Huffman-Codes, die aus Mengen von Musterdokumenten entwickelt wurden. Bezüglich dieser wurden sie optimiert (siehe Abbildungen)

Informationsgehalt der natürlichen Sprache

Wie viele Bit stecken im Mittel in einem Zeichen der Schriftsprache?

Naive Häufigkeitsanalyse auf Zeichenebene ergibt:
ca. 4 bit/Zeichen

Run-length	Terminating Codewords	
	White Runs	Black runs
0	001001	00001011
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
9	10100	000100
10	00111	0000100
11	01000	0000101
12	001000	0000111
13	000011	00000100
14	110100	00000111
15	110101	000011000
16	101010	000010111
17	101011	0000011000
18	0100111	0000001000
19	0001100	00001100111
20	0001000	00001101000
21	0010111	00001101100
22	0000011	00000110111
23	0000100	00000101000
24	0101000	0000010111
25	0101011	00000011000
26	1010011	000011001010
27	0101000	000011001011
28	0011000	000011001100
29	00000010	000011001101
30	00000011	000001101000
31	00011010	000001101001
32	00011111	0000011101010
33	00010010	000001101011
.	.	.
.	.	.
.	.	.
.	.	.

.
.
.
.
.
57	01011010	000001011000			
58	01011011	000001011001			
59	01001010	000000101011			
60	01001011	000000101100			
61	00110010	000001011010			
62	00110011	000001100110			
63	00110100	000001100111			
.
.
.
.
.
.
64	11011	0000001111			
128	10010	000011001000			
192	010111	000011001001			
256	0110111	000001011011			
320	00110110	000000110011			
384	00110111	000000110100			
448	01100100	000000110101			
512	01100101	0000001101100			
576	01101000	0000001101101			
640	01100111	0000001001010			
704	011001100	0000001001011			
768	011001101	0000001001100			
832	011010010	0000001001101			
896	011010011	0000001110010			
960	011010100	0000001110011			
1024	011010101	0000001110100			
1088	011010110	0000001110101			
1152	011010111	0000001110110			
1216	011011000	0000001110111			
1280	011011001	0000001010010			
1344	011011010	0000001010011			
1408	011011011	0000001010100			
1472	010011000	0000001010101			
1536	010011001	0000001010110			
1600	010011010	0000001010111			
1664	011000	0000001001000			
1728	010011011	0000001001001			
EOL	000000000001	000000000001			

Run Length (Black or White)	Make-up Codeword
1792	00000001000
1856	00000001100
1920	00000001101
1984	000000010010
2048	000000010011
2112	000000010100
2176	000000010101
2240	000000010110
2304	000000010111
2368	000000011100
2432	000000011101
2496	000000011110
2560	000000011111

Abbildung 4.6: Der Telefax-Code. Zunächst kommt ein *make-up*-Code, der die höherwertigen Bits der Lauflänge codiert, dann folgt – abhängig davon, ob ein schwarzes oder weißes Pixel-Intervall vorliegt (denn die Wahrscheinlichkeiten dafür sind verschieden!) – ein *terminating*-Code. Für „EOL“ (Zeilenende) gibt es einen speziellen, ausgezeichneten Code.

Abschätzung mit besserer Methode:

- Alphabet = Menge der Wörter
- Annahmen:
 - 20 Wörter (häufig) 25% aller Wörter im Text
 - 1000 Wörter 65% aller Wörter im Text
 - 40000(selten) 10 aller Wörter im Text

⇒ unterstelle Gleichwahrscheinlichkeit der Wörter innerhalb jeder der drei Gruppen

$$\begin{aligned}
 H(X) &:= \left(\sum_{i=1}^{20} \frac{0,25}{20} \log_2 \frac{20}{0,25} + \sum_{i=21}^{1020} \frac{0,65}{1000} \log_2 \frac{1000}{0,65} + \sum_{i=1021}^{41020} \frac{0,1}{40000} \log_2 \frac{40000}{0,1} \right) \text{ bit/Wort} \\
 &= 10,2 \text{ bit/Wort}
 \end{aligned}$$

Genauere Abschätzung mit **Häufigkeitswörterbuch**:¹

Wörter(n_i)	Anteil am Text	p_i	$-p_i \cdot n_i \cdot \text{ld}p_i$
3	10%	$\frac{10}{100 \cdot 3}$	0,491
63	40%	$\frac{40}{100 \cdot 63}$	2,920
254	22%	$\frac{22}{100 \cdot 254}$	2,238
80000	28%	$\frac{28}{100 \cdot 80000}$	5,075

$$\Rightarrow - \sum_{i=1}^4 p_i \cdot n_i \cdot \text{ld}p_i = 10,724 \text{ bit/Wort}$$

Wirklicher Wert ist kleiner, weil dieser Wert (gemäß stückweiser Gleichverteilung) eine obere Schranke ist.

$$10,2 \frac{\text{bit}}{\text{Wort}} / 7 \frac{\text{Zeichen}}{\text{Wort}} = 1,457 \frac{\text{bit}}{\text{Zeichen}}$$

Kann eine Codierung gefunden werden, die das realisiert?

Ja, ein optimaler Code mit $\bar{n} \leq H(X) + 1 = 11,2$

Objektiver, subjektiver und ästhetischer Informationsgehalt von Texten

Ratetest mit Text T : „subjektiver Informationsgehalt“

$$H_{\text{Experte}}(T) = H_{\text{Ästhet}}(T)$$

$$H_{\text{Laie}}(T) = H_{\text{sem}}(T) + H_{\text{Ästhet}}(T)$$

$$\begin{aligned} H_{\text{sem}}(T) &= \text{semantischer Informationsgehalt} \\ &= H_{\text{Laie}}(T) - H_{\text{Experte}}(T) \end{aligned}$$

¹nach Daten von Kaeding: „Häufigkeitswörterbuch der deutschen Sprache“

Versuche mit

mathematischen Texten:

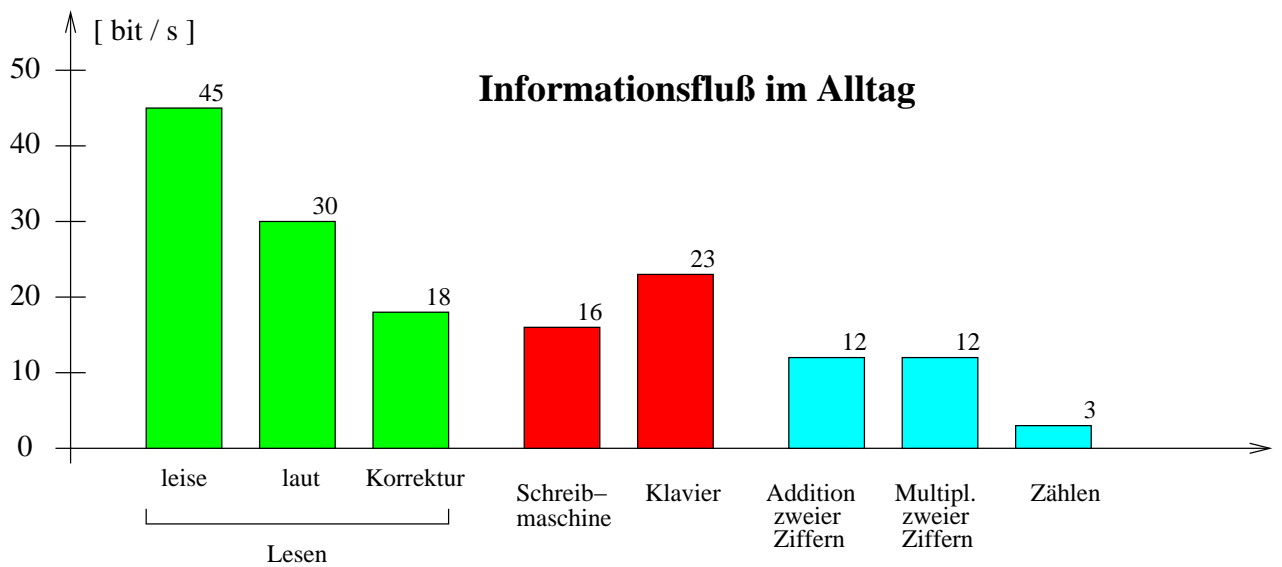
$$H_{\text{Ästhet}} \approx 0,5 \text{ Bit/Zeichen}$$

$$H_{\text{sem}} \approx 0,6 \text{ Bit/Zeichen}$$

literarischen Texten:

$$H_{\text{Ästhet}} \approx 1 \text{ bit/Zeichen}$$

Der Mensch als Verarbeiter der natürlichen Sprache



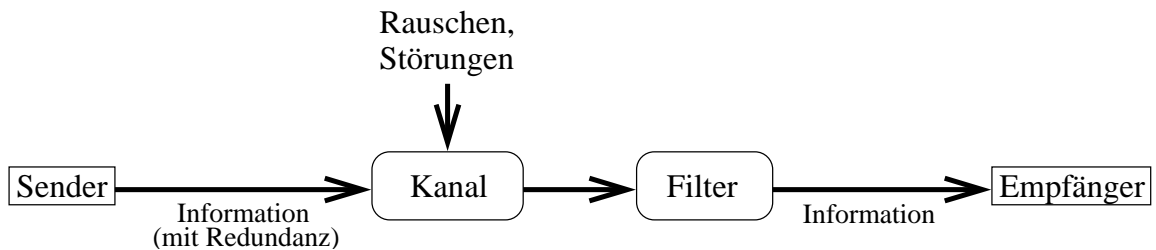
Sprechtempo: bis 600-800 Silben/Minute

Schreibmaschine: 900-1200 Anschläge/Minute

Schnelles lesen: schneller als Sprechen

Parlamentsstenograph: 400-500 Silben/Minute
⇒ 0,5 → 50 bit/Sekunde

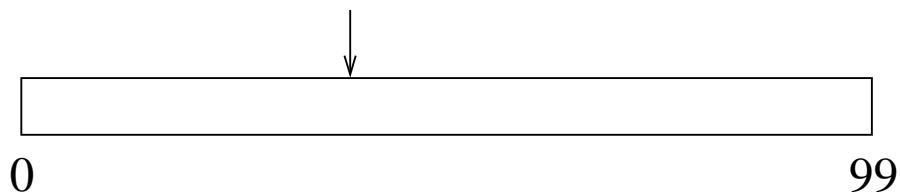
Erkennen von Information in Gegenwart von Rauschen



- Handschrift
- Sprechen
- Überforderung beim Textverstehen?
- Unterforderung beim Textverstehen?

Erkennungsleistung beim Kanalmodell

Ableseexperiment:



Input: Zeigerstellung zwischen 0...99

Output: Ableseergebnis zwischen 0...99, vom Menschen

Frage: *Wie viel Information pro Ablesung gewonnen?*

Kanal: $p(y_i|x_i)$ = bedingte Wahrscheinlichkeit, dass Zeigerstellung i eingestellt und j abgelesen ($i, j \in [0...99]$)

Wie groß ist die Kapazität des Kanals bzw. die obere Schranke dafür?

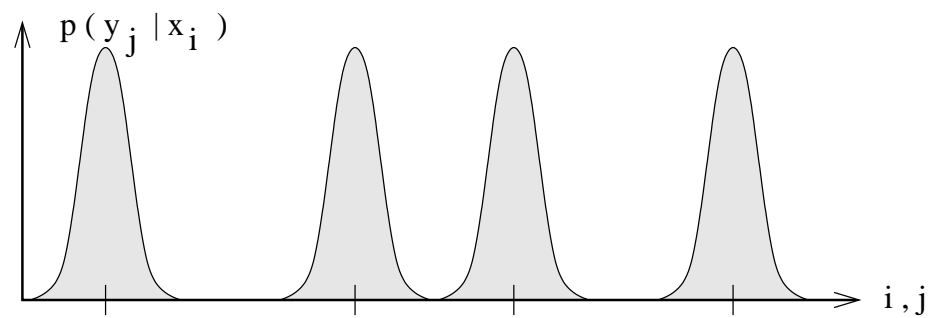


Abbildung 4.9: Näherungsrechnung für das Ableseexperiment

Anwendung der Informationstheorie auf die Dialoggestaltung

Vermeidung unnötiger Information:

- einheitlicher Bildschirmaufbau
- Bildschirm nicht überfrachten
- einheitliche Terminologie
- orthogonale Kommandostruktur
- einheitlicher Stil

Störungsfreier Kanal:

- hoher Kontrast
- scharfe Zeichenumrandung
- flackerfreies Bild