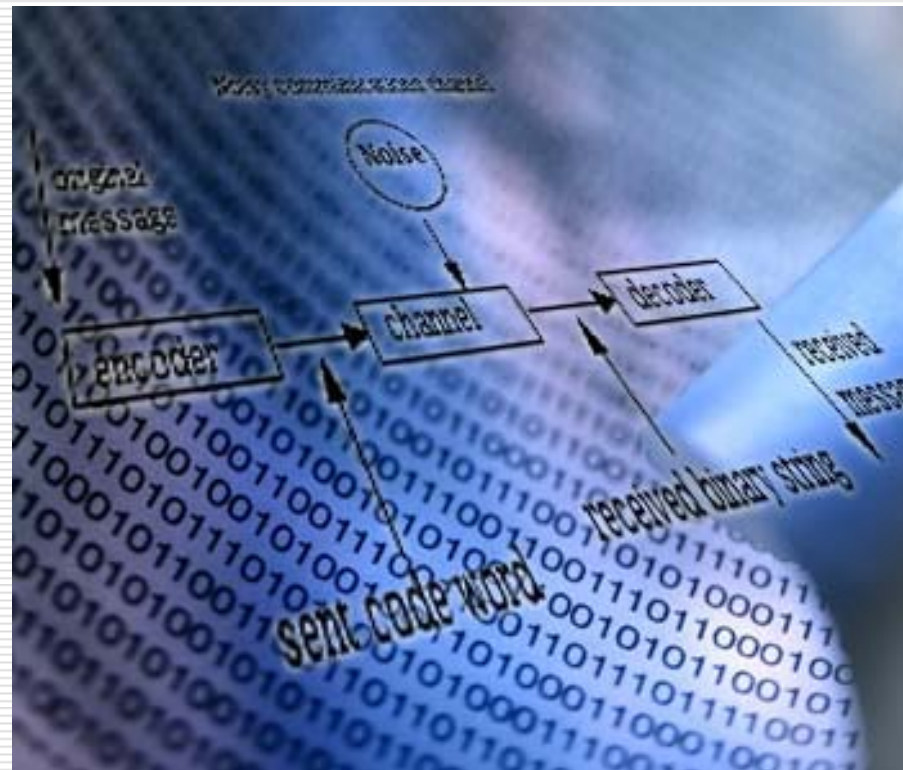


# Kapitel 9: Intervallängencodierung

---



# Ziele des Kapitels

---

- ❑ Universelle Datenkompression
- ❑ Codes für ganze Zahlen
- ❑ Intervalllängen-Codierung

- ❑ Problem bisher: statistische Eigenschaften der Quelle müssen bekannt sein
- ❑ Besser: Verfahren, welche sich auf die Eigenschaften der Quelle adaptieren
- ❑ **Definition:** Ein Verfahren heisst **universell** für eine gegebene Klasse von Informationsquellen, wenn es asymptotisch jede Quelle dieser Klasse auf die Entropierate codiert
- ❑ Asymptotisch bezieht sich hierbei auf die Wahl von Parametern, wie z.B. die Blocklänge

- Wir betrachten im Folgenden eine universelle Codierung für eine beliebige stationäre Quelle der Form  $X = X_1, X_2, \dots$
- Wir betrachten keine spezielle Zahlenstatistik
- Die Folge  $X$  wird in Blöcke der Länge  $L$  unterteilt
- Zunächst betrachten wir dazu eine präfixfreie Codierung für **ganze** Zahlen  $j$ , d.h.  $j \in \mathbb{N}^+$
- Die „normale“ binäre Codierung von  $j$ ,  $B(j)$  hat die Länge  $L(j)$ 
$$L(j) = \lfloor \log_2 j \rfloor + 1$$
- $B(j)$  ist nicht präfixfrei!

- Um einen präfixfreien Code  $C_1$  zu erhalten, können wir dem Codewort  $B(j)$  eine Folge von  $L(j)-1$  Symbolen „0“ voranstellen

$$C_1(j) = 0^{L(j)-1} \| B(j)$$

- Die resultierende Länge  $L_1(j)$  ist

$$L_1(j) = 2L(j) - 1 = 2 \lfloor \log_2 j \rfloor + 1$$

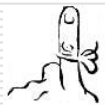
- Dieser Code ist nicht optimal
- Verbesserung durch Ersetzen der  $L(j)-1$  führenden „0“ mit präfixfreier Codierung von  $L(j)$ , also der Länge
- Dazu verwenden wir z.B.  $C_1(j)$

- Da jedes Codewort in  $B(j)$  immer mit einer „1“ beginnt, kann diese weggelassen werden
- Der resultierende Code sei  $B'$
- Wir erhalten als präfixfreie Codierung für die ganzen Zahlen somit

$$C_2(j) = C_1(L(j)) \| B'(j)$$

- Die Länge  $L_2(j)$  von  $C_2(j)$  ist also

$$L_2(j) = L_1(L(j)) + L(j) - 1 = \\ 2 \lfloor \log_2(\lfloor \log_2 j \rfloor + 1) \rfloor + \lfloor \log_2 j \rfloor + 1$$



Das obige Prinzip ist klar rekursiv und kann entsprechend verallgemeinert werden

# Code für ganze Zahlen

□  $j = 37$

→  $B(j) = 100101$

→  $L(37) = 6$

→  $C_1(37) = 0^5 \parallel B(j) = 00000100101$

→  $C_1(L(37)) = C_1(6) = 00110$

→  $L_1(37) = 11$

→  $C_2(37) = C_1(L(37)) \parallel B'(37)$   
=  $00110 \parallel 00101$   
=  $0011000101$

→  $L_2(37) = 10$

□ Lang, jedoch gut für grosse Zahlen  $L_2(10^6) = 28$

- ❑ Man verwende einen beliebigen, nicht degenerierten Code  $C$  für die Zahlen in  $\mathbb{N}^+$
- ❑ Man verwende einen präfixfreien Code für  $\mathbb{N}^+$ , um die Länge des Codewortes zu codieren
- ❑ Dieser Code wird dem Codewort vorangestellt
- ❑ Als trivialen präfixfreien Code für  $\mathbb{N}^+$  nehmen wir  $C'(j) = 0^j 1$
- ❑ Der so erhaltene präfixfreie Code kann wiederum zur Codierung der Längen verwendet werden
- ❑ Wir erhalten eine asymptotisch immer effizienter werdende Codierung der Zahlen  $j \in \mathbb{N}^+$



- Bei Verwendung der vorherigen Codes für  $B(j)$  und  $B'(j)$  erhalten wir folgende Rekursionsformel:

$$C^n(j) = \begin{cases} 0^{j-1} 1 & \text{für } n = 0 \\ C^{n-1}(L(j)) \| B'(j) & \text{für } n > 0 \end{cases}$$

- Diese ist nicht mehr von der Quellenstatistik abhängig
- Es zeigt sich, dass durch rekursive Anwendung asymptotisch optimale Codes entworfen werden können

- Eine sehr effiziente Codierung für stationäre binäre Quellen  $X = X_1, X_2, \dots$  mit starker Asymmetrie
- Es seien  $P_{X_i}(0) = 1 - p$      $P_{X_i}(1) = p$
- Für alle  $i \geq 1$  und  $p \ll 1/2$
- Eine lange Folge solcher Zufallsvariablen wird codiert, indem nur die Abstände aufeinanderfolgender „1“ Symbole codiert werden
- Man verwendet einen präfixfreien Code für ganze Zahlen
- Wir verwenden einfach  $C_2(j)$

- Wichtig für die folgende Analyse ist ein Code, bei dem gilt

$$\lim_{j \rightarrow \infty} \frac{L(j)}{\log_2 j} = 1$$

- Zur Abschätzung der Codewortlängen verwenden wir

$$L_2(j) \leq \tilde{L}_2(j) := 2 \log_2(\log_2 j + 1) + \log_2 j + 1$$

- Wir setzen weiterhin  $X_0 = 1$  als Initialisierung
- Wir berechnen den Erwartungswert des Abstandes zweier „1“-Symbole
- Der Abstand zwischen dem  $i$ -ten und  $i+1$ -tem „1“-Symbol sei  $D^{(i)}$

- Es ergibt sich für den Erwartungswert bis zum  $M$ -ten Auftreten einer „1“

$$E\left[\sum_{i=1}^M D^{(i)}\right] = \sum_{i=1}^M E[D^{(i)}] = M \cdot E[D]$$

- Dies gilt aufgrund der Stationarität, welche den Erwartungswert zeitkonstant hält
- Die relative Häufigkeit eines „1“-Symbols einer Folge von der Länge  $N$  ist

$$\frac{E\left[\sum_{i=1}^N X_i\right]}{N} = \frac{\sum_{i=1}^N E[X_i]}{N} = E[X_i] = p$$

- Somit gilt:

$$E[D] = \frac{1}{p}$$

- Die mittlere Distanz zwischen zwei „1“ Symbolen ist also reziprok zu deren Auftretenswahrscheinlichkeit
- $E[D]$  ist die mittlere Anzahl von Zufallsvariablen, die mit einem Codewort codiert werden
- Die mittlere Anzahl von Bits pro Codewort ist

$$E[L_2[D]] = \sum_{d=1}^{\infty} P_D(d) L_2(d)$$

- Die Funktion  $L_2$  ist konkav. Anwenden der Jensen-Ungleichung

$$E[L_2(D)] \leq E[\tilde{L}_2(D)] \leq \tilde{L}_2(E[D]) = \tilde{L}_2(1/p)$$

- Die mittlere Anzahl von Bits pro Zufallsvariable ist also

$$\frac{E[L_2(D)]}{E[D]} = p\tilde{L}_2(1/p) = 2p \log_2(1 - \log_2 p) - p \log_2 p + p$$

- Die Effizienz dieser Codierung ist für  $p$  gegen 0 optimal

- Wir erhalten 
$$\lim_{p \rightarrow 0} \frac{E[L_2(D)]}{E[D]} = h(p)$$

- ❑ Wir erweitern die Betrachtung auf nicht gedächtnisfreie,  $Q$ -äre Quellen
- ❑ Das Alphabet sei  $\{y_1, \dots, y_Q\}$
- ❑ Die Quelle sei  $\mathbf{Y} = Y_1, Y_2, \dots$  mit  $P_{Y_1} = P_{Y_2} = \dots = P_Y$
- ❑ Wesentlicher Unterschied: Jedes Symbol  $y$  muss bezüglich des letzten Auftretens des gleichen Symbols codiert werden
- ❑ Initialisierung, wie vorher, so dass
$$Y_i = y_{1-i} \quad \text{für} \quad i = -Q + 1, \dots, 0$$
- ❑ Sei  $D_i$  der Abstand zwischen dem Symbol an der  $i$ -ten Stelle ( $Y_i$ ) und seinem nächsten Auftreten

- Bei  $Y_i = y$  ist die Verteilung gleich wie im binären Fall, mit  $p = P_Y(y)$
- Voraussetzung: Alle Symbole wurden bereits einmal übertragen

- Es gilt 
$$E[D_i | Y_i = y] = \frac{1}{P_Y(y)}$$

- Anwendung der Jensen-Ungleichung

$$E[L_2(D_i) | Y_i = y] \leq E[\tilde{L}_2(D_i) | Y_i = y] \leq \tilde{L}_2(E[D_i | Y_i = y]) \leq \tilde{L}_2\left(\frac{1}{P_Y(y)}\right)$$

- Nun muss noch über alle Werte von  $y$  gemittelt werden

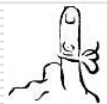


- Wir wenden erneut die Jensen-Ungleichung an

$$E[L_2(D_i)] = \sum_{y \in \mathcal{X}} P_Y(y) E[L_2(D_i) | Y_i = y] \leq \sum_{y \in \mathcal{X}} P_Y(y) \tilde{L}_2\left(\frac{1}{P_Y(y)}\right)$$

- Wir erhalten schliesslich

$$\begin{aligned} E[L_2(D_i)] &\leq H(Y) + 2 \log E\left[\log\left(\frac{1}{P_Y(y)}\right) + 1\right] + 1 \\ &= H(Y) + 2 \log(H(Y) + 1) + 1 \end{aligned}$$



Man sieht, dass diese Intervall-Längencodierung für grosse  $H(Y)$  nahezu asymptotisch optimal ist, wenn  $H(Y)$  zunimmt. Man nähert sich also der Quellenentropie  $H(Y)$ .

- ❑ Die beschriebene Methode lässt sich hinsichtlich Blocklängencodierung erweitern
- ❑ Wir unterteilen die Zeichenfolge der Quelle  $X$  in nicht-überlappende Blöcke der Länge  $L$
- ❑ Diese Folge von Blöcken stellt eine Ersatzquelle dar
- ❑ Wir wenden die Lauflängencodierung auf die neue Quelle an und erhalten

$$\lim_{p \rightarrow 0} \frac{E[L_2(D_i)]}{L} = H(X)$$



Es gibt keine universellen Kompressionsverfahren, die jede Quelle auf die Entropie komprimieren.

Die Behauptung, ein Programm komprimiere jeden Input auf  $x\%$  ( $0 < x < 100$ ), ist Unsinn.