ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Zusammenfassung Informationstheorie

Roman Böhringer

15. Juni 2020

Department of Computer Science, ETH Zürich

Inhaltsverzeichnis

Inhaltsverzeichnis **i**

1 Einführung **1**
1.1 Information und Kommunikation 1
1.2 Datenkompression- und -codierung 1

2 Grundlagen **3**
2.1 Shannon-Informationsgehalt 3
2.2 Entropie von Verteilungen / Zufallsvariablen 4
2.3 Gemeinsame / Bedingte Entropie 5
2.4 Wechselseitige Information 6
2.5 Kullback-Leibler-Divergenz 7

3 Quellcodierung **8**
3.1 Eigenschaften von Codes 8
3.2 Codebäume 9
3.3 Kraft-Ungleichung 9
3.4 Optimalität von Codes 11
3.5 Codes bzgl. falscher Verteilung 13
3.6 Huffman-Codes 13
3.6.1 Optimalität 14
3.7 Kraft-Ungleichung für eindeutig decodierbare Codes 15
3.8 Typische Ereignisse und Asymptotische Gleichverteilung . . . 16
3.8.1 Datenkompression 17
3.9 Arithmetische Codes 18
3.9.1 Performanz 20
3.9.2 Vergleich mit Huffmann-Codes 20
3.10 Lempel-Ziv-Codes 21

4 Kanalcodierung **22**

4.1 Kanalkapazität 22
4.1.1 Eigenschaften 24
4.2 Kanalcodierung 24
4.3 Shannons Kanalcodierungstheorem 27
4.3.1 Erreichbarkeit 27
4.3.2 Kapazität als obere Schranke 29
4.4 Fehlerbehaftete Kommunikation oberhalb der Kapazität . . . 31

5 Elementare Codierungstheorie **34**
5.1 Grundbegriffe 34
5.2 Lineare Codes 34
5.3 Generatormatrizen 35
5.4 Parity-Check-Matrizen 36
5.5 Hamming-Codes 36
5.6 Syndrom-Decodierung linearer Codes 37
5.7 Lineare Codes via Polynomevaluation 38
5.7.1 Erreichbare Distanzen 39
5.8 Reed-Solomon-Codes 39
5.8.1 Fehlerkorrektur 41
5.9 Polar-Codes 42

6 Anhang **47**
6.1 Jensen-Ungleichung 47

Kapitel 1

Einführung

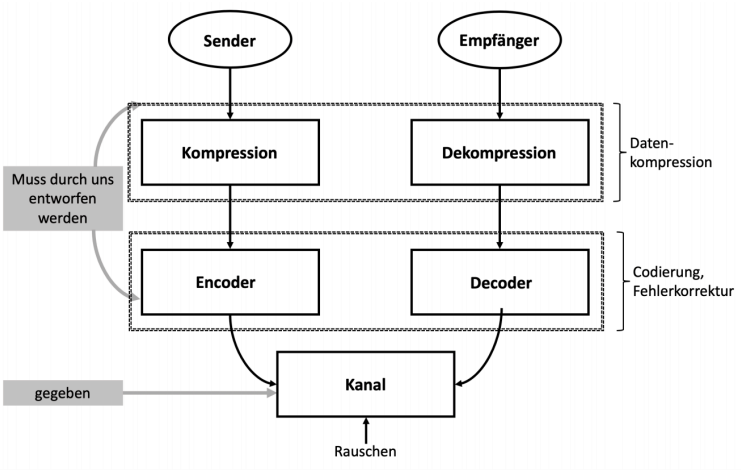
1.1 Information und Kommunikation

Eine Nachricht enthält Information, wenn durch sie die Unsicherheit reduziert wird. Der Informationsgehalt ist dabei ein relativer Begriff, er hängt vom Wissensstand beim Erhalt der Nachricht ab.

Kommunikation bedeutet die Übertragung von Information von A nach B über einen Kanal, wobei A und B verschiedene Orte (mit bspw. einer Telefonleitung als Kanal) oder Zeiten sein können (mit bspw. einem Speichermedium als Kanal).

1.2 Datenkompression- und -codierung

Das allgemeine Kommunikationsmodell ist nachfolgend dargestellt.



Kompression und Codierung sind nötig wegen der Begrenztheit von Ressourcen (Zeit, Bandbreite, Speicher, etc...) und der Unvollkommenheit von Kanälen (”Rauschen”).

Die Informationstheorie befasst sich mit den folgenden Fragen:

1. Wie stark kann man Daten (annähernd) verlustfrei komprimieren? (Entfernen von Redundanz, Entropie)
2. Wie kann man Daten so codieren / decodieren, dass ein verrauschter Kanal möglichst effizient genutzt wird und die Wahrscheinlichkeit fehlerhafter Übertragung gering ist? (kontrolliertes Hinzufügen von Redundanz, Kanalkapazität)

Kapitel 2

Grundlagen

2.1 Shannon-Informationsgehalt

Sei (Ω, P) ein diskreter Wahrscheinlichkeitsraum. Jedem Ereignis $A \in 2^\Omega$ soll eine Zahl zugeordnet werden, welche misst, wieviel Information wir erhalten, wenn wir erfahren, dass $A \in 2^\Omega$ eingetreten ist. Die Funktion sollte erfüllen:

1. $h(\Omega) = 0$
2. $A \subset B \Rightarrow h(A) \geq h(B)$
3. $h(A \cap B) = h(A) + h(B)$ falls A, B unabhängig
4. $P(A) = P(B) \Rightarrow h(A) = h(B)$

Aus 4. ergibt sich, dass h die Gestalt

$$h(A) = g(P(A))$$

haben muss. Wegen 3. muss $g(p_0 p_1) = g(p_0) + g(p_1)$ gelten. Somit haben wir:

$$\begin{aligned} g(p_0 p_1) &= g(p_0) + g(p_1) && \text{(Eigenschaft 4)} \\ p_1 g'(p_0 p_1) &= g'(p0) && \text{(Nach } p_0 \text{ ableiten)} \\ g'(p_0 p_1) + p_0 p_1 g''(p_0 p_1) &= 0 && \text{(Nach } p_1 \text{ ableiten)} \\ g'(u) + g''(u) &= 0 && (u = p_0 p_1 \text{ einführen)} \end{aligned}$$

Diese Differenzialgleichung führt zu $g(u) = k \log u$ für $k \in \mathbb{R}$. Aus 2. folgt $k < 0$, womit bis auf einen konstanten Faktor gilt:

$$g(p) = -\log p = \log \frac{1}{p}$$

Für ein Ereignis $A \in 2^\Omega$ mit $P(A) > 0$ ist

$$h(A) = -\log P(A)$$

der (Shannon-)Informationsgehalt von A.

2.2 Entropie von Verteilungen / Zufallsvariablen

Die Entropie einer Verteilung $P : \Omega \to [0,1]$ ist:

$$H(P) := -\sum_{\omega \in \Omega} P(\omega) \log P(\omega)$$

Die Entropie einer Zufallsvariable $X : \Omega \to \mathcal{X}$ mit Verteilung P_X ist $H(X) := H(P_X)$, also

$$H(X) := -\sum_{x \in X(\Omega)} P_X(x) \log P_X(x)$$

Die Entropie wird in Bits angegeben.

Die Entropie kann als Erwartungswert der Zufallsvariablen

$$\begin{aligned} h : X(\Omega) &\to \mathbb{R}_{\geq 0} \\ x &\mapsto h(x) = -\log P_X(x) \end{aligned}$$

angesehen werden, d.h. $H(X) = E(h)$. $h(x)$ ist der Informationsgehalt des Ereignisses $X = x$, somit kann $H(X)$ als "durchschnittliche Information, die wir durch Bekanntwerden des Wertes von X erhalten", interpretiert werden.

Für jede Zufallsvariable $X : \Omega \to \mathcal{X}$ gilt $0 \leq H(X) \leq \log |X(\Omega)|$ mit

$$H(X) = 0 \Leftrightarrow \exists x \in \mathcal{X} : P_X(x) = 1$$

$$H(X) = \log |X(\Omega)| \Leftrightarrow P_X(x) = \frac{1}{|X(\Omega)|} \quad \forall x \in X(\Omega)$$

Beweis: Die untere Schranke $0 \leq H(X)$ ergibt sich, weil $[0,1] \to \mathbb{R}, p \mapsto -p \log p$ nichtnegativ ist. Für die obere Schranke wird die Jensen-Ungleichung auf die konkave Funktion $[0,1] \to \mathbb{R}, p \mapsto \log p$ angewandt:

$$H(X) = E[-\log P_X] = E[\log(\frac{1}{P_X})] \leq \log E[\frac{1}{P_X}] = \log |X(\Omega)|$$

(da $E[\frac{1}{P_X}] = \sum_{x \in X(\Omega)} P_X \frac{1}{P_X} = |X(\Omega)|$). Damit die untere Schranke angenommen wird, müssen alle Wahrscheinlichkeiten gleich 0 oder 1 sein, da $-p \log p > 0$ für $p \in (0,1)$. Da $p \mapsto \log p$ strikt konkav ist, folgt aus der Jensen-Ungleichung, dass die obere Schranke $H(X) = \log |X(\Omega)|$ nur angenommen werden kann, wenn alle Wahrscheinlichkeiten gleich sind.

Eine zweite Interpretation der Entropie ist, dass sie die "Unsicherheit über den Wert von X, bevor wir ihn erfahren", misst. $H(X)$ ist maximal, wenn a priori alle Werte gleich wahrscheinlich sind und minimal, wenn nur ein Wert möglich.

2.3 Gemeinsame / Bedingte Entropie

Die gemeinsame Entropie von zwei Zufallsvariablen $X : \Omega \to \mathcal{X}$ und $Y : \Omega \to \mathcal{Y}$ mit gemeinsamer Verteilung P_{XY} ist

$$H(X,Y) := -\sum_{x,y} P_{X,Y}(x,y) \log P_{X,Y}(x,y)$$

$H(X,Y)$ ist also die Entropie der Zufallsvariablen $(X,Y) : \Omega \to \mathcal{X} \times \mathcal{Y}$

Für $y \in Y(\Omega)$ mit $P_Y(y) > 0$ existiert die auf $\{Y = y\}$ bedingte Verteilung $P_{X|Y=y}$ mit

$$P_{X|Y=y} = P(X = x|Y = y) = \frac{P_{X,Y}(x,y)}{P_Y(y)}$$

Die Entropie dieser Verteilung ist

$$H(X|\ Y = y) = -\sum_x P_{X|Y=y}(x) \log P_{X|Y=y}(x)$$

und misst die "Unsicherheit über den Wert von X, wenn $Y = y$ bekannt ist".

Die bedingte Entropie von X gegeben Y ist definiert als:

$$H(X|Y) = \sum_y P_Y(y) H(X|Y = y)$$

$H(X|Y)$ ist somit der Erwartungswert von $H(X|Y = y)$ bzgl. der möglichen Werte von Y. Es gilt:

$$\begin{aligned} H(X|Y) &= \sum_y P_Y(y) \left(-\sum_x P_{X|Y=y}(x) \log P_{X|Y=y}(x) \right) \\ &\overset{(*)}{=} -\sum_{x,y} P_{X,Y}(x,y) \left[\log P_{X,Y}(x,y) - \log P_Y(y) \right] \\ &= -\sum_{x,y} P_{X,Y}(x,y) \log P_{X,Y}(x,y) + \sum_y \left(\sum_x P_{X,Y}(x,y) \right) \log P_Y(y) \\ &\overset{(**)}{=} H(X,Y) - H(Y) \end{aligned}$$

Wobei $(*)$ aus der Definition von $P_{X|Y=y}$ folgt und $(**)$, weil $\sum_x P_{X,Y}(x,y) = P_Y(y)$.

Somit folgt die "Kettenregel", die besagt, dass für jedes Paar von Zufallsvariablen $X : \Omega \to \mathcal{X}, Y : \Omega \to \mathcal{Y}$ gilt:

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

Daraus folgt, dass $H(X) \leq H(X,Y)$ mit Gleichheit genau dann, wenn Y eindeutig durch X bestimmt ist. *Beweis:* $H(X) \leq H(X,Y)$ folgt sofort aus

$H(X) = H(X,Y) - H(Y|X)$ (da die Entropie positiv ist). Für $H(X) = H(X,Y)$ muss $H(Y|X) = 0$ gelten, also muss $H(Y|X = x) = 0$ für alle x gelten. Wir haben in Abschnitt 2.2 bereits gezeigt (mit der Jensen-Ungleichung), dass dies genau der Fall ist, wenn $P_{y|X=x} = 1$ für ein y, d.h. Y eindeutig durch X bestimmt ist.

2.4 Wechselseitige Information

$H(X|Y)$ ist die "Unsicherheit über X, wenn Y bekannt ist". Die Grösse

$$I(X;Y) := H(X) - H(X|Y)$$

ist demnach die "Reduktion von Unsicherheit über X durch Bekanntwerden von Y"(durchschnittlich bzgl. Y) und wird wechselseitige Information von X und Y genannt. Aus der Kettenregel folgt:

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

Somit ist $I(X;Y)$ symmetrisch in X und Y, d.h. $I(X;Y) = I(Y;X)$.

Für alle X, Y gilt $I(X;Y) \geq 0$ mit Gleichheit genau dann, wenn X und Y unabhängig sind. *Beweis:*

$$\begin{aligned}
I(X;Y) &= H(X) + H(Y) - H(X,Y) \\
&= -\sum_x P_X(x) \log P_X(x) - \sum_y P_Y(y) \log P_Y(y) + \sum_{x,y} P_{X,Y}(x,y) \log P_{X,Y}(x,y) \\
&= -\sum_{x,y} P_{X,Y}(x,y) \left(\log P_X(x) + \log P_Y(y)\right) + \sum_{x,y} P_{X,Y}(x,y) \log P_{X,Y}(x,y) \\
&= -\sum_{x,y} P_{X,Y}(x,y) \log \frac{P_X(x)P_Y(y)}{P_{X,Y}(x,y)} \\
&= \mathbb{E}_{P_{X,Y}} \left[-\log \frac{P_X(x)P_Y(y)}{P_{X,Y}(x,y)} \right]
\end{aligned}$$

$z \mapsto -\log z$ ist strikt konvex, womit mit der Jensen-Ungleichung folgt:

$$\begin{aligned}
\mathbb{E}\left[-\log \frac{P_X(x)P_Y(y)}{P_{XY}(x,y)} \right] &\geq -\log \left(\mathbb{E}\left[\frac{P_X(x)P_Y(y)}{P_{XY}(x,y)} \right] \right) \\
&= -\log \left(\sum_{x,y} P_{XY}(x,y) \frac{P_X(x)P_Y(y)}{P_{XY}(x,y)} \right) \\
&= 0
\end{aligned}$$

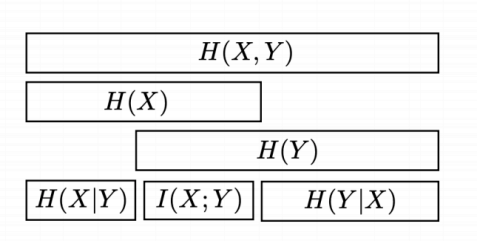
Und Gleichheit gilt genau dann, wenn $\frac{P_X(x)P_Y(y)}{P_{XY}(x,y)} = 1$ für alle $x \in \mathcal{X}, y \in \mathcal{Y}$.

Aus $I(X;Y) = H(X) - H(X|Y)$ und somit $H(X) = I(X;Y) + H(X|Y)$ folgt somit direkt, dass Bedingen die Entropie vermindert, d.h.:

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

Mit Gleichheit genau dann, wenn X und Y unabhängig sind.

Die Beziehung zwischen den Entropiegrössen ist nachfolgend dargestellt:



2.5 Kullback-Leibler-Divergenz

Die wechselseitige Information $I(X;Y)$ kann als Abweichung der Zufallsvariablen X, Y davon, unabhängig zu sein, interpretiert werden. Die Kullback-Leibler-Divergenz misst die "Distanz" zwischen zwei Verteilungen $P, Q : \Omega \to [0,1]$ und ist definiert als:

$$D_{KL}(P||Q) := -\sum_{\omega} P(\omega) \log\left(\frac{Q(\omega)}{P(\omega)}\right) = E_P\left[\log\frac{P}{Q}\right]$$

D_{KL} ist nicht symmetrisch, womit $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ im Allgemeinen gilt. Somit ist die Kullback-Leibler-Divergenz insbesondere keine Metrik (wofür Symmetrie eine Anforderung ist).

Die Gibbs-Ungleichung besagt, dass $D_{KL}(P||Q) \geq 0$ gilt, mit Gleichheit genau dann, wenn $P = Q$ ist. *Beweis:* Da $z \mapsto -\log z$ strikt konvex ist, gilt:

$$\begin{aligned} D_{KL}(P||Q) &= E_P\left[-\log\frac{Q}{P}\right] \\ &\geq -\log E_P[\frac{Q}{P}] \\ &= 0 \end{aligned}$$

Für Gleichheit muss $-\log\frac{Q}{P}$ konstant sein und da P und Q Verteilungen sind, folgt somit, dass diese Konstante 1 sein muss, d.h. $P = Q$.

Kapitel 3

Quellcodierung

Die Grundfrage bei der Quellcodierung ist, wie die von einer Datenquelle emittierte Information möglichst kompakt als Binärstring (o.ä.) dargestellt werden kann.

Ein Quellcode (oder einfach Code) über einem Codealphabet $\mathcal{D}$ für eine Menge $\mathcal{X}$ ist eine Abbildung $C : \mathcal{X} \to \mathcal{D}^*$. $C(x)$ ist das Codewort für $x \in \mathcal{X}$ und $l_C(x)$ die Länge von $C(x)$. Ein Beispiel ist der Morse-Code, welcher $\mathcal{X} = \{A, B, \ldots, Z\}$ auf $\mathcal{D}^* = \{\cdot, -\}^*$ abbildet mit $C(A) = \cdot, C(X) = \cdot - \cdot\cdot, \ldots$.

Ein Code für eine Zufallsvariable $X : \Omega \to \mathcal{X}$ ist ein Code für $X(\Omega)$, d.h. eine Abbildung $C : X(\Omega) \to \mathcal{D}^*$. Die erwartete Länge des Codes ist:

$$L_C = \sum_{x \in X(\Omega)} P_X(x) l_C(x)$$

d.h. $L_C = E_{P_X}[l_C]$. Die Wahl von $C : X(\Omega) \to \mathcal{D}^*$, so dass L_C minimal wird, hängt von der Verteilung ab.

3.1 Eigenschaften von Codes

- Ein Code heisst nicht-degeneriert, wenn $C : \mathcal{X} \to \mathcal{D}^*$ eine injektive Abbildung ist, d.h.

  $$x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j) \quad \forall i \neq j$$

- Jeder Code $C : \mathcal{X} \to \mathcal{D}^*$ kann erweitert werden zu einem Code $C^* : \mathcal{X}^* \to \mathcal{D}^*$, gegeben durch:

  $$C^*(x_1, \ldots, x_n) \mapsto C(x_1) \ldots C(x_n)$$

  wobei $(x_1, \ldots, x_n)$ und $C(x_1) \ldots C(x_n)$ die Verkettung der x_i / $C(x_i)$ sind.

- C wird eindeutig decodierbar genannt, wenn C^* nicht-degeneriert ist, d.h.:

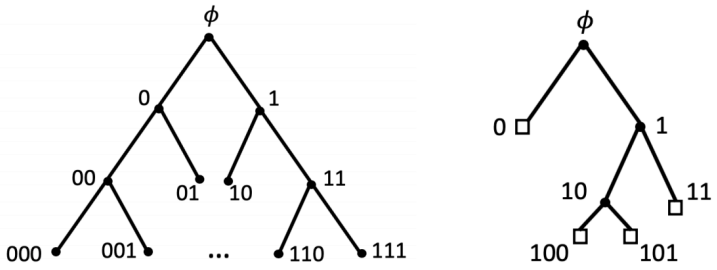
$$x_1 \ldots x_m \neq x'_1 \ldots x'_n \Rightarrow C(x_1) \ldots C(x_m) \neq C(x'_1) \ldots C(x'_n)$$

- C wird präfixfrei (instantan) genannt, wenn kein Codewort $C(x)$ Präfix eines anderen Codeworts $C(y)$ ist.

Jeder präfixfreie Code ist eindeutig decodierbar (aber nicht jeder eindeutig decodierbare Code ist präfixfrei). Der Vorteil von präfixfreien Codes ist, dass beim Decodieren das Ende eines Wortes sofort erkannt werden kann, ohne das Decodieren weiterer Symbole abzuwarten.

3.2 Codebäume

$\{0,1\}^*$ kann als Menge der Knoten eines unendlichen Binärbaumes aufgefasst werden. Für einen Code $C : \mathcal{X} \to \{0,1\}^*$ entspricht die Menge der Codewörter $C(\mathcal{X}) \subset \{0,1\}^*$ einer Teilmenge der Knoten dieses Baums. Ein Wort $x \in \{0,1\}^*$ ist genau dann ein Präfix eines Wort $y \in \{0,1\}^*$, wenn der y entsprechende Knoten Nachfahre des x entsprechenden Knotens ist. Somit ist ein Code C genau dann präfixfrei, wenn die C entsprechenden Knoten die Blätter eines Teilbaums sind:



3.3 Kraft-Ungleichung

1. Für jeden präfixfreien Code $C : \mathcal{X} \to \{0,1\}^*$ gilt:

$$\sum_{x \in \mathcal{X}} 2^{-l_C(x)} \leq 1$$

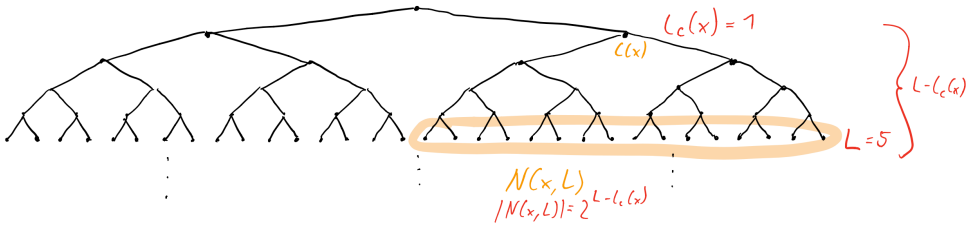
2. Falls umgekehrt eine Funktion $l : \mathcal{X} \to \mathbb{N}$ die Ungleichung

$$\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq 1$$

erfüllt, dann existiert ein präfixfreier Code $C : \mathcal{X} \mapsto \{0,1\}^*$, so dass $l_C(x) = l(x)$ für alle $x \in \mathcal{X}$ gilt.

Beweis: **1.)** Für $l \in \mathbb{N}$ sei $\nu_C(l)$ die Zahl der Codewörter in $C(\mathcal{X})$ der Länge l. Wir fixieren ein $L \in \mathbb{N}$ und betrachten ein Codewort $C(x)$ der Länge $l_C(x) < L$. Sei $N(x,L) \subset \{0,1\}^L$ die Menge derjenigen Wörter der Länge L, die Nachfahren von $C(x)$ sind und daher nicht Codewörter sein können. Es gilt:

$$|N(x,L)| = 2^{L-l_C(x)}$$



Für Codewörter $C(x), C(x')$ mit $x \neq x'$ sind $N(x,L)$ und $N(x',L)$ disjunkt. Aus diesem Grund können wir schliessen:

$$\left| \bigcup_{x, l_C(x)<L} N(x,L) \right| = \sum_{x, l_C(x)<L} 2^{L-l_C(x)} \leq 2^L - \nu_C(L)$$

(die Ungleichung folgt, da wir total 2^L Strings der Länge L haben, wovon $\nu_C(L)$ Codewörter sind. Summieren wir nun die Anzahl der Strings mit Länge L, die kein Codewort sein können (Betrag der Vereinigung der $N(x,L)$) und addieren die Strings, die Codewörter sind (also $\nu_C(L)$), muss dies kleiner als 2^L sein).

Es gilt ausserdem:

$$\sum_{x, l_C(x)<L} 2^{L-l_C(x)} = \sum_{l=1}^{L-1} \nu_C(l) 2^{L-l} = 2^L \sum_{l=1}^{L-1} \nu_C(l) 2^{-l} = 2^L \sum_{l=1}^{L} \nu_C(l) 2^{-l} - \nu_C(L)$$

Ersetzen wir nun $\sum_{x, l_C(x)<L} 2^{L-l_C(x)}$ durch diesen Ausdruck in obiger Ungleichung, folgt:

$$\sum_{l=1}^{L} \nu_C(l) 2^{-l} \leq 1$$

Dies gilt für jedes $L \in \mathbb{N}$, somit also:

$$\sum_{l=1}^{\infty} \nu_C(l) 2^{-l} \leq 1 \Leftrightarrow \sum_{x \in \mathcal{X}} 2^{-l_C(x)} \leq 1$$

2.) Beweis via Induktion: Wir nehmen an, dass wir für alle x mit $l(x) < L$ bereits Codewörter konstruiert haben, so dass der Code präfixfrei ist und die vorgegebenen Wortlängen hat. Aus $\sum_{x\in\mathcal{X}} 2^{-l(x)} \leq 1$ folgt durch Umkehrung der Schritte aus dem Beweis von 1.), dass

$$\sum_{x,l(x)<L} 2^{L-l(x)} \leq 2^L - \nu(L)$$

Dabei ist $\nu(L) := |\{x \in \mathcal{X} | l(x) = L\}|$ die Anzahl der Wörter, die durch die Funktion auf auf die Länge L gemapped werden. Die linke Seite der Ungleichung ist die Anzahl der Wörter der Länge L, die ein bereits gewähltes Codewort als Präfix hat. Da es insgesamt 2^L Wörter der Länge L gibt, können wir daraus also schliessen, dass $\geq \nu(L)$ Wörter der Länge L existieren, die keines der bereits gewählten Codewörter als Präfix haben und somit als Codewort verwendet werden können. Wenn eine solche Funktion l gegeben ist, kann also ein präfixfreier Code konstruiert werden.

3.4 Optimalität von Codes

Sei X eine Zufallsvariable und $p_i := P_X(x_i)$, wobei $X(\Omega) := \{x_i | i \in I\}$. Ein Code $C^* : X(\Omega) \to \{0,1\}^*$ wird optimal genannt, wenn für jeden anderen Code C für X gilt: $L_{C^*} \leq L_C$.

Um optimale Werte für die Codewortlängen $l_i = l_C(x_i)$ zu finden, muss $\sum_i p_i l_i$ minimiert werden. Aus der Kraft-Ungleichung folgt die Bedingung $\sum_i 2^{-l_i} \leq 1$ mit der Bedingung $\sum_i 2^{-l_i} = 1$. $\sum_i p_i l_i$ kann mittels Lagrange-Multiplikatoren minimiert werden:

$$\mathcal{L} = \sum_i p_i l_i + \lambda \sum_i 2^{-l_i}$$
$$\frac{\partial \mathcal{L}}{\partial l_i} = p_i - \lambda \ln 2 * 2^{-l_i}$$

Dies ist genau dann null, wenn $2^{-l_i^*} = \frac{p_i}{\lambda \ln 2}$ und wegen $\sum_i 2^{-l_i^*} = 1 = \sum_i p_i$ folgt $\lambda = \frac{1}{\ln 2}$ und somit $2^{-l_i^*} = p_i$ bzw.

$$l_i^* = -\log p_i$$

Die erwartete Länge des Codes ist dann:

$$\sum_i p_i l_i^* = -\sum_i p_i \log p_i = H(X)$$

Da i.A. $l_i \notin \mathbb{N}$ gibt es keinen Code, der diese Länge realisiert.

Die erwartete Länge L_C eines präfixfreien Codes $C : X(\Omega) \to \{0,1\}^*$ für eine Zufallsvariable X erfüllt

$$L_C \geq H(X)$$

mit Gleichheit genau dann, wenn die Codewortlängen $2^{-l_i} = p_i$ erfüllen.
Beweis:

$$\begin{aligned} L_C - H(X) &= \sum_i p_i l_i + \sum_i p_i \log p_i \\ &= -\sum p_i \log 2^{-l_i} + \sum_i p_i \log p_i \end{aligned}$$

Einführen der Verteilung $q_i := \frac{2^{-l_i}}{Z}$ mit $Z := \sum_j 2^{-l_j}$ ergibt:

$$\begin{aligned} &= -\textstyle\sum_i p_i \log (q_i Z) + \sum_i p_i \log p_i \\ &= -\textstyle\sum_i p_i \log \left(\frac{q_i}{p_i}\right) - \log Z \\ &= D_{KL}(p\|q) - \log Z \\ &\geq 0 \end{aligned}$$

Da $Z = \sum_j 2^{-l_j} \leq 1$ (Kraft-Ungleichung). Für Gleichheit muss $D_{KL}(p\|q) = 0$ und $Z = 1$, also $p_i = 2^{-l_i}$ gelten.

Wenn $-\log p_i \in \mathbb{N}$ nicht gilt, setzen wir $l_i := \lceil -\log p_i \rceil$, womit $\sum_i 2^{-l_i} \leq \sum_i 2^{\log p_i} = \sum_i p_i = 1$ gilt, die Kraft-Ungleichung also erfüllt ist. Für die Länge dieses Codes $\hat{C} : X(\Omega) \to \{0,1\}^*$ gilt:

$$L_{\hat{C}} = \sum_i p_i l_i < \sum_i p_i(-\log p_i + 1) = H(X) + 1$$

Somit folgt, dass jeder optimale präfixfreie Code $C^* : X(\Omega) \to \{0,1\}^*$ für X folgende Ungleichung erfüllt:

$$H(X) \leq L_{C^*} < H(X) + 1$$

(wobei die untere Schranke $H(X)$ aus dem vorherigen Beweis folgt und die obere Schranke $H(X) + 1$ aus der Konstruktion von $\hat{C}$).

Durch das Codieren von längeren Sequenzen, kann die obere Schranke an die erwartete Länge pro Symbol verbessert werden. Dabei werden Blöcke von der Quelle X als Zufallsvariable modelliert:

$$X^N = (X_1, \ldots, X_N) : \Omega \to X(\Omega) \times \ldots X(\Omega)$$

mit i.i.d. X_i. Wir haben dann für $H(X^N) = NH(X)$. Gemäss dem oberen Theorem existiert ein Code C_N für X^N mit

$$NH(X) = H(X^N) \leq L_{C_N} < H(X^N) + 1 = NH(X) + 1$$

Die erwartete Länge des Codes pro Symbol $\frac{1}{N} L_{C_N}$ ist also kleiner als $H(X) + \frac{1}{N}$ und es gilt $\frac{1}{N} L_{C_N} \xrightarrow{N\to\infty} H(X)$.

3.5 Codes bzgl. falscher Verteilung

Was passiert, wenn wir $q(x_i)$ als Verteilung von X annehmen und einen Code C_q mit Codewortlängen $l_{C_q}(x_i) = \lceil -\log q(x_i) \rceil$ verwenden? Die bzgl. p erwartete Länge $E_p[l_{C_q}]$ erfüllt dann:

$$H(X) + D_{KL}(p\|q) \leq E_p[l_{C_q}] < H(X) + D_{KL}(p\|q) + 1$$

Beweis: Aus $-\log q(x) \leq \lceil -\log q(x) \rceil < -\log q(x) + 1$ folgt:

$$\begin{aligned} E_p[l_{C_q}] \geq -\sum_x p(x) \log q(x) &= -\sum_x p(x)(\log q(x) + \log p(x) - \log p(x)) \\ &= H(X) + D_{KL}(p\|q) \end{aligned}$$

Und:

$$\begin{aligned} E_p[l_{C_q}] < -\sum_x p(x) \log q(x) + 1 &= -\sum_x p(x)(\log q(x) + \log p(x) - \log p(x)) + 1 \\ &= H(X) + D_{KL}(p\|q) + 1 \end{aligned}$$

3.6 Huffman-Codes

Huffman-Codes liefern einen einfachen Algorithmus für die Konstruktion optimaler präfixfreier Codes für X. Die Idee ist, einen Codebaum rekursiv von den Blättern her aufzubauen. Die Verteilung von X ist gegeben durch $\mathbf{p} = (p_1, \ldots, p_n)$ mit $p_i := P_X(x_i)$, $i = 1, \ldots, n$. Ein Huffmann-Code (HC) für $\mathbf{p}$ wird so bestimmt:

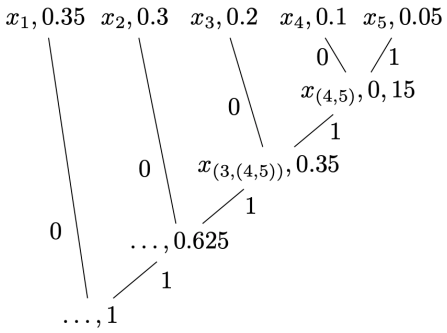
1. Wähle zwei Worte x_{i_0}, x_{i_1} mit kleinsten Wahrscheinlichkeiten p_{i_0}, p_{i_1} aus, d.h. so, dass $p_{i_0} + p_{i_1} \leq p_j + p_k \quad \forall j \neq k = 1, \ldots, n$.
2. Konstruiere einen HC C'_H für die Verteilung p' auf $\{x_1, \ldots, x_n, x_{(i_0,i_1)}\} \backslash \{x_{i_0}, x_{i_1}\}$ mit $p'_{(i_0,i_1)} = p_{i_0} + p_{i_1}$ und $p'_j = p_j$ für $j \neq (i_0, i_1)$
3. Definiere:
$$C_H(x_j) = \begin{cases} C'_H(x_{(i_0,i_1)})0 & \text{falls } j = i_0 \\ C'_H(x_{(i_0,i_1)})1 & \text{falls } j = i_1 \\ C'_H(x_j) & \text{sonst} \end{cases}$$

Die Zahl der Wörter wird bei jedem Schritt um 1 reduziert, bei nur noch zwei Wörtern x_0, x_1 ist der Code $x_0 \mapsto 0, x_1 \mapsto 1$.

Das Verfahren kann als Baum dargestellt werden:

X	v	w	x	y	z
P_X	0.35	0.3	0.2	0.1	0.05
C_H	0	10	110	1110	1111

Tabelle 2: Verteilung einer Zufallsvariablen X und zugehöriger Huffman-Code.



3.6.1 Optimalität

Sei C_H ein Huffman-Code für eine Zufallsvariable X und sei C^* ein optimaler präfixfreier Code für X. Dann gilt:

$$L_{C_H} = L_{C^*}$$

C_H ist also ebenfalls ein optimaler Code für X. *Beweis:* Mittels vollständiger Induktion über $n = |X(\Omega)|$, wobei die Aussage für $n = 1, 2$ klar ist. Angenommen, sie ist für $n = 1, \ldots, N-1$ und beliebige Verteilungen bewiesen. Wir schreiben $L_H(p)$ / $L^*(p)$ für die erwartete Länge eines HC / optimalen präfixfreien Codes für die Verteilung $\mathbf{p} = (p_1, \ldots, p_n)$ Seien i_0 und i_1 so, dass $p_{i_0} + p_{i_1} \leq p_j + p_k \quad \forall j, k = 1, \ldots, n$ und sei $\mathbf{p}'$ gegeben durch $p'_{i_0,i_1} = p_{i_0} + p_{i_1}$ und $p'_j = p_j$ für $j \neq i_0, i_1$ (wie beim HC-Algorithmus). Dann gilt:

$$\begin{aligned} L_H(\mathbf{p}) &= L_H(\mathbf{p}') + p_{i_0} + p_{i_1} \\ &= L^*(\mathbf{p}') + p_{i_0} + p_{i_1} \end{aligned}$$

(wobei die Gleichheit für die zweite Zeile aus der Induktionsannahme folgt). Zu zeigen ist $L^*(\mathbf{p}') \leq L^*(\mathbf{p}) - p_{i_0} - p_{i_1}$, woraus dann $L_H(\mathbf{p}) \leq L^*(\mathbf{p})$ mit obiger Gleichung direkt folgt und aus der Optimalität von L^* somit $L_H(\mathbf{p}) = L^*(\mathbf{p})$.

Sei also C_p^* ein optimaler Code für $\mathbf{p}$. Unter den am wenigsten wahrscheinlichsten Codewörtern von C_p^* gibt es zwei, die sich nur im letzten Bit unterscheiden (*), weswegen wir annehmen können, dass sich $C_p^*(x_{i_0})$ und $C_p^*(x_{i_1})$ nur im letzten Bit unterscheiden. Somit kann ein Code $C_{p'}$ für $\mathbf{p}'$ definiert werden, indem $C_{p'}(x_j) = C_p^*(x_j)$ für $j \neq i_0, i_1$ gesetzt wird und $C_{p'}(x_{i_0,i_1})$ als das längste gemeinsame Präfix von $C_p^*(x_{i_1})$ und $C_p^*(x_{i_2})$ definiert wird

(mit Länge $|C_p^*(x_{i_1})| - 1$). Somit gilt $L_{C_{p'}} = L^*(\mathbf{p}) - p_{i_0} - p_{i_1}$ und es folgt $L^*(\mathbf{p}') \leq L_{C_{p'}} = L^*(\mathbf{p}) - p_{i_0} - p_{i_1}$ (da $L^*(\mathbf{p}')$ die Länge eines optimalen Codes ist, ist sie sicher kleiner / gleich als die von $L_{C_{p'}}$).

Beweis (*): Sei C' ein optimaler präfixfreier Code für X. Es ist klar, dass es für jedes Codewort w_0 mit maximaler Länge ein weiteres Codewort w_1 geben muss, welches sich nur im letzten Bit unterscheidet (ansonsten könnte man w_0/w_1 um ein Bit kürzen, was der Optimalität widersprechen würde). Ausserdem muss ein Paar $x_{i_0} \neq x_{i_1}$ mit kleinster Wahrscheinlichkeit existieren, so dass $C'(x_{i_0})$ und $C'(x_{i_1})$ maximale Länge haben. Ansonsten könnte man einen kürzeren Code durch Vertauschen der Codewörter (Paar mit kleinster Wahrscheinlichkeit zu längsten Codewörtern) erhalten, was ein Widerspruch zur Optimalität wäre. Somit kann C' durch Vertauschen von Codewörtern mit maximaler Länge (wodurch sich die Länge nicht ändert) so modifiziert werden, dass x_{i_0} auf w_0 und x_{i_1} auf w_1 abgebildet werden, womit wir einen optimalen Code mit der geforderten Eigenschaft erhalten.

3.7 Kraft-Ungleichung für eindeutig decodierbare Codes

Jeder eindeutig decodierbare Code $C : \mathcal{X} \to \{0,1\}^*$ erfüllt die Kraftungleichung, d.h. $\sum_{x\in\mathcal{X}} 2^{-l_C(x)} \leq 1$.

Beweis: Wir betrachten C^k, die k-te Erweiterung von C (k-fache Verkettung von C), welche per Definition nicht-degeneriert ist und somit maximal 2^n Codewörter der Länge n enthält. Für die Länge eines Codewortes gilt:

$$l(x_1, \ldots, x_n) = \sum_{i=1}^{k} l(x_i)$$

Wir wollen $\sum_{x\in\mathcal{X}} 2^{-l(x)} \leq 1$ zeigen und betrachten dafür die k-te Potenz:

$$\begin{aligned}\left(\sum_{x\in\mathcal{X}} 2^{-l(x)}\right)^k &= \sum_{x_1\in\mathcal{X}} \sum_{x_2\in\mathcal{X}} \cdots \sum_{x_k\in\mathcal{X}} 2^{-l(x_1)} 2^{-l(x_2)} \ldots 2^{-l(x_k)} \\ &= \sum_{x_1,x_2,\ldots,x_k\in\mathcal{X}^k} 2^{-l(x_1)} 2^{-l(x_2)} \ldots 2^{-l(x_k)} \\ &= \sum_{x^k\in\mathcal{X}^k} 2^{-l(x^k)}\end{aligned}$$

Wir führen nun l_{max} als maximale Codeword Länge ein und $a(m)$ als Anzahl der Quellwörter x^k, die einem Codewort der Länge m zugeordnet werden. Aus der eindeutigen Decodierbarkeit folgt $a(m) \leq 2^m$. Wir haben:

$$\sum_{x^k\in\mathcal{X}^k} 2^{-l(x^k)} = \sum_{m=1}^{kl_{max}} a(m) 2^{-m}$$

Somit:

$$\left(\sum_{x\in\mathcal{X}} 2^{-l(x)}\right)^k = \sum_{m=1}^{kl_{max}} a(m)2^{-m} \leq \sum_{m=1}^{kl_{max}} 2^m 2^{-m} = kl_{max}$$

Also $\sum_j 2^{-l_j} \leq (kl_{max})^{1/k}$ Dies gilt für alle k und somit auch für $k \to \infty$, womit $\sum_j 2^{-l_j} \leq 1$ folgt.

3.8 Typische Ereignisse und Asymptotische Gleichverteilung

Die asymptotische Gleichverteilungseigenschaft (AEP) besagt: Falls die Zufallsvariablen $X_1, X_2, \ldots : \Omega \to \mathcal{X}$ unabhängig sind und die gleiche Verteilung wie $X : \Omega \to \mathcal{X}$ haben, dann gilt für jedes $\varepsilon > 0$:

$$P\left(|-\frac{1}{N}\ \log P(X_1,\ldots,X_N) - H(X)| > \varepsilon\right) \to 0 \text{ für } N \to \infty$$

(Konvergenz von $-\frac{1}{N}\ \log P(X_1,\ldots,X_N) \to H(X)$ in Wahrscheinlichkeit) *Beweis:* Das schwache Gesetz der grossen Zahlen wird auf die Zufallsvariable $-\log p(X_i)$ angewandt:

$$-\frac{1}{N}\log P(X_1,\ldots,X_N) = -\frac{1}{N}\sum_{i=1}^{N}\log P(X_i) \xrightarrow{N\to\infty} E[-\log P(X)] = H(X)$$

Für gegebene ε, N wird die typische Menge definiert:

$$\begin{aligned} T_{N,\varepsilon} &:= \left\{\mathbf{x} \in \mathcal{X}^N || -\frac{1}{N}\log P(\mathbf{x}) - H(X)| \leq \varepsilon\right\} \\ &= \left\{\mathbf{x} \in \mathcal{X}^N | H(X) - \varepsilon < -\frac{1}{N}\log P(\mathbf{x}) < H(X) + \varepsilon\right\} \\ &= \left\{\mathbf{x} \in \mathcal{X}^N | 2^{-N(H(X)+\varepsilon)} < P(\mathbf{x}) < 2^{-N(H(X)-\varepsilon)}\right\} \end{aligned}$$

Eigenschaften der typischen Menge sind:

1. $P(T_{N,\varepsilon}) \to 1$ für $N \to \infty$ (AEP)
2. $|T_{N,\varepsilon}| < 2^{N(H(X)+\varepsilon)}$
3. $|T_{N,\varepsilon}| > (1-\delta)2^{N(H(X)-\varepsilon)}$ für jedes $\delta \in (0,1)$ und jedes $N \gg 1$

1. folgt direkt aus dem AEP. 2. folgt, weil:

$$1 = \sum_{\mathbf{x}\in\mathcal{X}^n} P(\mathbf{x}) \geq \sum_{\mathbf{x}\in T_{N,\varepsilon}} P(\mathbf{x}) > \sum_{\mathbf{x}\in T_{N,\varepsilon}} 2^{-N(H(X)+\varepsilon)} = 2^{-N(H(X)+\varepsilon)}|T_{N,\varepsilon}|$$

3. folgt auch aus dem AEP, da

$$1-\delta < P(T_{N,\varepsilon}) < \sum_{\mathbf{x}\in T_{N,\varepsilon}} 2^{-N(H(X)-\varepsilon)} = |T_{N,\varepsilon}|2^{-N(H(X)-\varepsilon)}$$

3.8.1 Datenkompression

Ein Blockcode $C : \mathcal{X}^N \to \{0,1\}^*$ wird wie folgt definiert: Zuerst werden zwei eindeutig decodierbare Codes definiert:

- $T_{N,\varepsilon} \to \{0,1\}^{\lceil N(H(X)+\varepsilon)\rceil}$ (wegen $|T_{N,\varepsilon}| < 2^{N(H(X)+\varepsilon)}$ möglich)
- $\mathcal{X}^N \backslash T_{N,\varepsilon} \to \{0,1\}^{N\log|\mathcal{X}|}$

Diese können zu $C : \mathcal{X}^N \to \{0,1\}^*$ kombiniert werden, indem jedem Codewort das Präfix 0 (falls $\mathbf{x} \in T_{N,\varepsilon}$) bzw. 1 (falls $\mathbf{x} \in \mathcal{X}^N \backslash T_{N,\varepsilon}$) vorangestellt wird. C ist eindeutig decodierbar und es gilt:

$$L_C \leq P\left(T_{N,\varepsilon}\right)\left(N(H(X)+\varepsilon)+2\right) + \left(1-P\left(T_{N,\varepsilon}\right)\right)\left(N\log|\mathcal{X}|+2\right)$$

Wird N nun so gross gewählt, dass $P(T_{N,\varepsilon}) > 1-\varepsilon$, erhält man:

$$\begin{aligned} L_C &\leq N(H(X)+\varepsilon)+2+\varepsilon(N\log|\mathcal{X}|+2) \\ &= N\left(H(X)+\varepsilon'\right) \end{aligned}$$

Mit $\varepsilon' = \varepsilon + \frac{2}{N} + \varepsilon\log|\mathcal{X}| + 2\varepsilon$, weswegen die Codelänge pro Symbol ca. $H(X)$ ist für grosses N.

Die Frage ist nun, ob die von einer Quelle X emittierte Information auf weniger als $H(X)$ Bits pro Symbol komprimiert werden kann, wenn eine Fehlerwahrscheinlichkeit $\delta > 0$ in Kauf genommen wird. Die Idee dabei ist, eine Menge $S_\delta \subset \mathcal{X}^N$ mit $P(S_\delta) \geq 1-\delta$ zu finden, wobei $|S_\delta|$ möglichst klein sein soll. Dann wird ein Code $S_\delta \to \{0,1\}^*$ konstruiert und $\mathcal{X}^N \backslash S_\delta$ auf einen "Fehlerstring" abgebildet, womit die Wahrscheinlichkeit, dass der Code für ein Wort der Länge N nicht eindeutig decodiert werden kann, $< \delta$ ist. Für $\delta \in (0,1)$ wird definiert:

$$H_\delta(X^N) := \min\{\log|S_\delta| \,|\, S_\delta \subset \mathcal{X}^N, P(S_\delta) \geq 1-\delta\}$$

D.h. der Logarithmus der Mächtigkeit der kleinsten Menge, so dass die Wahrscheinlichkeit $\geq 1-\delta$ ist.

Shannon's Quellcodierungstheorem besagt, dass es für alle $\varepsilon > 0$ und $\delta \in (0,1)$ ein $N_0 \in \mathbb{N}$ gibt, so dass für alle $N > N_0$ folgendes gilt:

$$\left|\frac{1}{N}H_\delta(X^N) - H(X)\right| < \varepsilon$$

Diese Ungleichung besagt:

1. $\frac{1}{N}H_\delta(X^N) < H(X) + \varepsilon$, es werden also ca. $H(X)$ Bits pro Symbol benötigt (was bereits bekannt ist).
2. $H(X) - \varepsilon < \frac{1}{N}H_\delta(X^N)$, d.h. selbst für grosse δ (hohe Fehlerwahrscheinlichkeiten) werden ca. $H(X)$ Bits benötigt.

Beweis: **1.)** N_0 wird so bestimmt, dass $P(T_{N,\varepsilon}) \geq 1 - \delta$ für $N \geq N_0$. Aus $|T_{N,\varepsilon}| < 2^{N(H(X)+\varepsilon)}$ folgt $\log|T_{N,\varepsilon}| < N(H(X) + \varepsilon)$ und aus $H_\delta(X^N) \leq \log|T_{N,\varepsilon}|$ (per Definition von $H_\delta(X^N)$ als Minimum) folgt

$$\frac{1}{N}H_\delta(X^N) < H(X) + \varepsilon$$

2.) Für ein gegebenes N wird eine Menge $S \subset \mathcal{X}^N$ betrachtet, so dass $\frac{1}{N}\log|S| \leq H(X) - \varepsilon$ und somit $|S| \leq 2^{N(H(X)-\varepsilon)}$. Dann gilt:

$$\begin{aligned} P(S) &= P\left(S \cap T_{N,\frac{\varepsilon}{2}}\right) + P\left(S \cap T^C_{N,\frac{\varepsilon}{2}}\right) \leq \sum_{x \in S} P(T_{N,\frac{\varepsilon}{2}}) + P\left(S \cap T^C_{N,\frac{\varepsilon}{2}}\right) \\ &\leq 2^{N(H(X)-\varepsilon)}2^{-N\left(H(X)-\frac{\varepsilon}{2}\right)} + P\left(T^C_{N,\frac{\varepsilon}{2}}\right) \\ &= 2^{-N\frac{\varepsilon}{2}} + P\left(T^C_{N,\frac{\varepsilon}{2}}\right) \overset{N\to\infty}{\longrightarrow} 0 \end{aligned}$$

Da somit für S mit $\frac{1}{N}\log|S| \leq H(X) - \varepsilon$ gilt, dass $P(S) < 1 - \delta$ (für $N \geq N_0$) folgt daraus, dass $\frac{1}{N}H_\delta(X^N)$ sicher grösser als $H(X) - \varepsilon$ sein muss (da für jedes kleinere $|S|$ gemäss obiger Argumentation $P(S) < 1 - \delta$, was $H_\delta(X^N)$ widerspricht).

3.9 Arithmetische Codes

Arithmetische Codes weisen den Strings $X_1, \ldots, X_N$ ein Intervall $I \subset [0, 1)$ zu, das durch einen binären String codiert wird.

Wir nehmen an, dass wir ein probabilistisches Modell der Quelle haben:

$$P(X_N = s_i | X_1, \ldots, X_{N-1})$$

wobei $X_N \in \mathcal{X} = \{s_1, \ldots, s_M\}$ das N-te emittierte Symbol ist. Um das Intervall für $X_1, \ldots, X_N$ zu finden, wird $[0, 1)$ in M Teilintervalle $I_1, \ldots, I_M$ unterteilt, wobei Intervall I_m Länge $P(X_1 = a_m)$ hat. Jedes der Teilintervalle I_m wird in Teilintervalle $I_{m,m'}$ unterteilt mit einer Länge proportional zu $P(X_2 = a_{m'}|X_1 = a_m)$ usw... Der String $a_{i_1}, \ldots, a_{i_N}$ wird dann dem Intervall $I_{i_1,\ldots,i_N}$ zugeordnet.

Beispiel 3. *Sei* $\mathcal{X} = \{a, b, c\}$ *mit* $P(X_i = a) = 0.5$, $P(X_i = b) = 0.3$ *und* $P(X_i = c) = 0.2$ *mit unabhängigen* X_i. *Abbildung 8 zeigt, welche Intervalle verschiedenen Strings zugeordnet werden.*

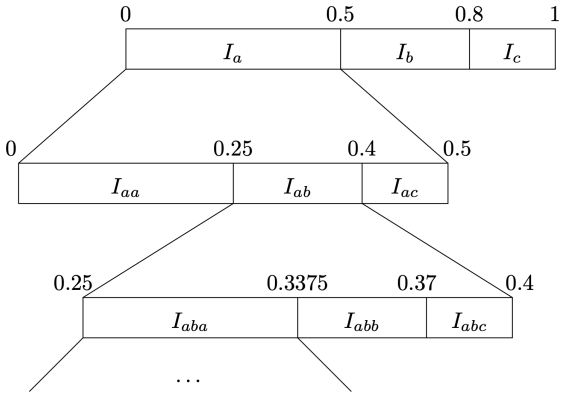


Abbildung 8: Subintervalle zur Konstruktion arithmetischer Codes.

Für $\mathbf{x}, \mathbf{x}' \in \{s_1, \ldots, s_m\}^*$ gilt, dass $I_{x'} \subseteq I_x$ genau dann, wenn $\mathbf{x}$ ein Präfix von $\mathbf{x}'$ ist. Wir nehmen an, dass das Alphabet ein spezielles "end-of-file" Symbol $s_M = \square$ besitzt. Wörter der Form $\mathbf{x} = s_{i_1} \ldots s_{i_l}$ mit $s_{i_1} \ldots s_{i_{l-1}} \neq \square$ und s_{i_l} sind dann eindeutig durch die Angabe irgendeiner Zahl $c \in I_x$ bestimmt.

Um davon ausgehend einen binären Code zu erhalten, werden Intervalle der Form

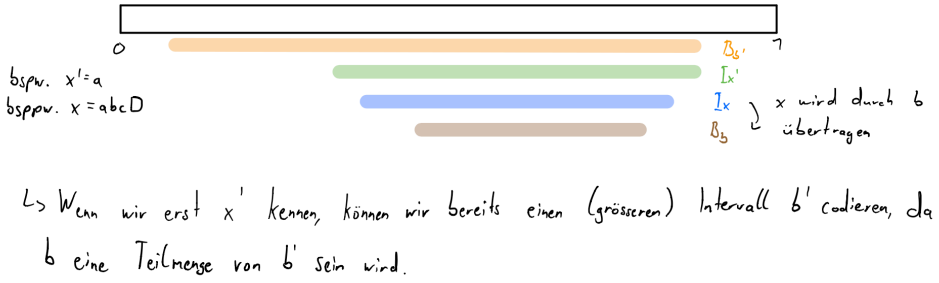
$$\left[\sum_{i=1}^{N} b_i 2^{-i}, \quad 2^{-N} + \sum_{i=1}^{N} b_i 2^{-i} \right), \quad b_i \in \{0, 1\}$$

durch $b_1 \ldots b_N \in \{0,1\}^*$ beschrieben. Zur Codierung von $\mathbf{x} = s_{i_1}, \ldots, s_{i_l}$ mit $s_{i_1}, \ldots, s_{i_{l-1}} \neq \square$ und $s_{i_l} = \square$ wird der kürzeste Binärstring angegeben, dessen Intervall $B_{b_1,\ldots,b_N}$ ganz in $I_{\mathbf{x}}$ liegt.

$\mathbf{x}$ kann codiert werden, bevor es ganz bekannt ist. Dies ist möglich, da für $\mathbf{x}, \mathbf{x}' \in \{s_1, \ldots, s_M\}^*$ und $\mathbf{b}, \mathbf{b}' \in \{0,1\}^*$ gilt mit:

$$I_{\mathbf{x}'} \subset B_{\mathbf{b}'} \text{ und } B_{\mathbf{b}} \subset I_{\mathbf{x}}$$

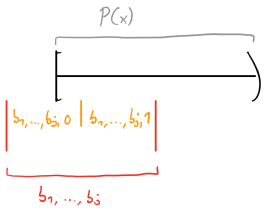
Falls $\mathbf{x}'$ Präfix von $\mathbf{x}$ ist, ist $\mathbf{b}'$ Präfix von $\mathbf{b}$. *Beweis:* Ist $\mathbf{x}'$ Präfix von $\mathbf{x}$, dann gilt $I_{\mathbf{x}} \subset I_{\mathbf{x}'}$. Aus $B_{\mathbf{b}} \subset I_{\mathbf{x}} \subset I_{\mathbf{x}'} \subset B_{\mathbf{b}'}$ folgt somit $B_{\mathbf{b}} \subset B_{\mathbf{b}'}$, d.h. $\mathbf{b}'$ ist ein Präfix von $\mathbf{b}$.



↳ Wenn wir erst x' kennen, können wir bereits einen (grösseren) Intervall b' codieren, da b eine Teilmenge von b' sein wird.

3.9.1 Performanz

Das Intervall I_x eines Wortes $x \in \{s_1, \ldots, s_M\}^*$ hat die Länge $P(x)$. Mit $j := \lceil -\log P(x) \rceil$ gilt $2^{-j} \leq P(x)$ und es gibt ein $B_{b_1 \ldots b_{j+1}} \subseteq I_x$ der Beschreibungslänge $j+1$.



Damit erfüllt die Länge des Codeworts die Ungleichung

$$l(x) \leq j + 1 \leq -\log P(x) + 2$$

Falls $X_1, X_2, \ldots$ unabhängig sind, gilt

$$\frac{1}{N} E_P[l(X_1, \ldots, X_N)] \leq -\frac{1}{N} E_p \left[\sum_{i=1}^{N} \log P(X_i) \right] + \frac{2}{N} \xrightarrow{N \to \infty} H(X)$$

3.9.2 Vergleich mit Huffmann-Codes

Huffmann-Codes sind Symbol-Codes, d.h. die Codierung einer Nachricht geschieht durch Verkettung der Codewörter. Die Wahrscheinlichkeiten der Symbole müssen vor der Codierung bekannt sein und die erwartete Länge ist bis zu 1 Bit über der Entropie $H(X)$. Bei der Codierung von Blöcken ist sie pro Symbol nur $\frac{1}{N}$ über der Entropie, aber die Berechnung von $|\mathcal{X}|^N$ Wahrscheinlichkeiten ist nötig.

Arithmetische Codes sind "Stream-Codes", d.h. es wird nicht einfach verkettet. Die Wahrscheinlichkeiten können beim Codieren gelernt und angepasst werden. Die erwartete Länge pro Symbol ist asymptotisch immer $H(X)$.

3.10 Lempel-Ziv-Codes

Die Grundidee ist beim Codieren ein "Wörterbuch"mit bisher aufgetretenen Phrasen aufzubauen, es existiert kein explizites probabilistisches Modell (wie bei arithmetischen Codes).

Beim Codieren eines Wortes wird das Wort zuerst unterteilt, indem ein "|" nach jeder bisher nicht gesehenen Phrase gesetzt wird, wobei eine neue Phrase beim letzten "|" beginnt. Jede dieser Phrasen wird in einem "Wörterbuch" D_n, das anfangs nur das leere Wort ($D_0 = \emptyset$) enthält und $n+1$ Wörter nach dem n-ten "|". Für jedes D_n wird der binäre Code $D_n \to \{0,1\}^{\lceil \log(n+1) \rceil}$ verwendet, der die $n+1$ Phrasen in D_n gemäss Auftretensreihenfolge auf die ersten $n+1$ Wörter in $\{0,1\}^{\lceil \log(n+1) \rceil}$ abbildet. Die $(n+1)$-te Phrase ist bis auf das letzte Symbol gleich einem Wort aus D_n (per Definition) und sie wird durch die Angabe dieses Worts und einem Code $\{A, B\} \to \{0,1\}$ für das letzte Symbol codiert.

$$
\begin{aligned}
&A|AB|ABB|B|ABA|ABBA|\ldots \\
&,0|1,1|10,1|00,1|010,0|\ldots
\end{aligned}
$$

so dass

$$
\begin{aligned}
D_0 &= \{\emptyset\} \\
D_1 &= \{\emptyset, A\} && \equiv \{0,1\} \\
D_2 &= \{\emptyset, A, AB\} && \equiv \{00,01,10\} \\
D_3 &= \{\emptyset, A, AB, ABB\} && \equiv \{00,01,10,11\} \\
D_4 &= \{\emptyset, A, AB, ABB, B\} && \equiv \{000,001,010,011,100\} \\
D_5 &= \{\emptyset, A, AB, ABB, B, ABA\} && \equiv \{000,001,010,011,100,101\} \\
&\ldots
\end{aligned}
$$

Der Decodierer muss das Wörterbuch rekonstruieren. Dazu setzt er zunächst die "|", wobei der erste nach einem Bit, dann nach 2 Bits, dann zwei nach je 3 Bits oder im Allgemeinen 2^{k-1} Phrasen der Länge $k+1$ (mit Ausnahme der ersten Phrase der Länge 1).

Die Lempel-Ziv-Codierung ist asymptotisch optimal, muss aber das Wörterbuch speichern.

Kapitel 4

Kanalcodierung

Ein diskreter, gedächtnisloser Kanal besteht aus einem Eingabealphabet $\mathcal{X}$, einem Ausgabealphabet $\mathcal{Y}$ und einer Familie $(P(\cdot|x))_{x\in\mathcal{X}}$ von Verteilungen auf $\mathcal{Y}$. Dabei ist $P(y|x)$ die "Wahrscheinlichkeit, y zu empfangen, wenn x gesendet wurde". Die durch $(P(y|x))_{x,y}$ gegebene Matrix wird auch Übergangsmatrix genannt. Der Kanal wird als gedächtnislos bezeichnet, weil die Ausgabeverteilung nur vom gerade gesendeten x abhängt, d.h. $P(y_n|x_1,\dots,x_n,y_1,\dots,y_{n-1}) = P(y_n|x_n)$

Ein Beispiel ist der binäre symmetrische Kanal (BSC) mit $\mathcal{X} = \mathcal{Y} = \{0,1\}$ mit $P(y=0|x=0) = P(y=1|x=1) = 1-p$ und $P(y=0|x=1) = P(y=1|x=0) = p$.

Wir sagen grundsätzlich, dass "viel Information" übertragen wird, wenn

1. Der durchschnittliche Informationsgehalt der Eingabe (d.h. die Entropie $H(X)$) gross ist.
2. Bei gegebener Ausgabe die durchschnittliche Unsicherheit bzgl. der Eingabe (d.h. $H(X|Y)$) möglichst klein ist.

Aufgrund von $H(X) - H(X|Y) = I(X;Y)$ legt dies nahe, dass $I(X;Y)$ möglichst gross sein sollte. Bei gegebenem Kanal hängt $I(X;Y)$ nur von P_X ab, da $P_{Y|X}$ gegeben ist.

4.1 Kanalkapazität

Die Kapazität eines Kanals ist

$$C = \max_{P_X} I(X;Y)$$

wobei über alle Eingabeverteilungen P_X maximiert wird.

Für den BSC gilt bspw. für jede Verteilung P_X der Eingabe:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(X) - \sum_y P(y) H(X|Y=y) \\ &= H(X) - \sum_y P(y) H(p, 1-p) \\ &\leq 1 - H(p, 1-p) \end{aligned}$$

Das Maximum wird angenommen, wenn X gleichverteilt ist, wobei dann $C = 1 - H(p, 1-p)$ ist.

Für einen Kanal mit nicht überlappender Ausgabe kann die Eingabe durch die Ausgabe eindeutig rekonstruiert werden (d.h. $P(X = x|Y = y) \in \{0, 1\}$ für alle x, y) und die Kapazität ist 1 (erreicht durch die Gleichverteilung $P_X = (\frac{1}{2}, \frac{1}{2})$).

Beim binären Auslöschungskanal ist das Eingabealphabet $\mathcal{X} = \{0, 1\}$ und das Ausgabealphabet $\mathcal{Y} = \{0, 1, ?\}$. Jedes gesendete Bit wird mit Wahrscheinlichkeit p gelöscht (in ein "?" verwandelt) und der Empfänger weiss, welche Bits gelöscht wurden. Es gilt:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(X) - \sum_y P(y) H(X|Y=y) \\ &= H(X) - P(Y=?) H(X|Y=?) \\ &= (1-p) H(X) \\ &\leq 1 - p \end{aligned}$$

Gleichheit wird auch hier durch die Gleichverteilung $P_X = (\frac{1}{2}, \frac{1}{2})$ erreicht.

Die verrauschte Schreibmaschine (noisy typewriter) hat ein Eingabe- und Ausgabealphabet $\mathcal{X} = \{A, \ldots, Z, \square\} = \mathcal{Y}$ und eine Übergangsmatrix $P(Y = \square|X = A) = P(Y = A|X = A) = P(Y = B|X = A) = \frac{1}{3}$, $P(Y = A|X = B) = P(Y = B|X = B) = P(Y = C|X = B) = \frac{1}{3}$, etc... Für die Kapazität gilt:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - \sum_x P_X(x) H(Y|X=x) \\ &\leq \log 27 - \log 3 \\ &= \log 9 \end{aligned}$$

Die Schranke wird (unter anderem) durch die Gleichverteilung $P_X(x) = \frac{1}{27}$ erreicht.

4.1.1 Eigenschaften

- $C \geq 0$, da $I(X;Y) \geq 0$
- $C \leq \log|\mathcal{X}|$, da $C = \max_{P_x} I(X;Y) \leq \max_{P_X} H(X) = \log|\mathcal{X}|$
- $C \leq \log|\mathcal{Y}|$ (analog)

Bei gegebenem $\mathcal{X}$ ist die Menge der möglichen Verteilungen P_X kompakt und $I(X;Y)$ als Funktion von P_X stetig, weswegen $C = \max_{P_x} I(X;Y)$ angenommen wird (d.h. es existiert eine Verteilung P_X, mit welcher die Kapazität erreicht wird).

4.2 Kanalcodierung

Gegeben sei ein Kanal $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$. Sei P_X eine Verteilung über $\mathcal{X}$ und P_Y die Verteilung über $\mathcal{Y}$ ($P_Y(y) = \sum_x P_X(x) P_{Y|X}(x,y)$). Wir betrachten Blöcke $\mathbf{x} \in \mathcal{X}^N$ für $N \gg 1$ und eine typische Menge $T^X_{N,\varepsilon} = \left\{\mathbf{x} \in \mathcal{X}^N \middle| \left| -\frac{1}{N}\log P_X(\mathbf{x}) - H(X)\right| < \varepsilon\right\}$ für ein kleines $\varepsilon > 0$. Für ein gegebenes $\mathbf{x} \in T^X_{N,\varepsilon}$ gibt es dann ca. $2^{NH(Y|X)}$ Blöcke $\mathbf{y} \in \mathcal{Y}^N$, die typisch bzgl. $P_{Y^N|X^N=\mathbf{x}}$ sind, d.h. typischerweise auftreten, wenn $\mathbf{x}$ über den Kanal gesendet wird. Da für alle typischen $\mathbf{x}, \mathbf{x}'$ gilt $P(\mathbf{x}) \approx P(\mathbf{x}')$, ist $H(Y^N|X^N = \mathbf{x}) \approx H(Y^N|X^N)$.

Die Grundidee ist, die typische Menge $T^Y_{N,\varepsilon}$ in nicht überlappende Teilmengen der Grösse $2^{NH(X|Y)}$ aufzuteilen, die jeweils einem $x \in \mathcal{X}^N$ entsprechen. Wegen $|T^Y_{N,\varepsilon}| \approx 2^{NH(Y)}$ gibt es maximal:

$$\frac{2^{NH(Y)}}{2^{NH(Y|X)}} = 2^{NH(Y)-NH(Y|X)} = 2^{NI(X;Y)}$$

solcher Teilmengen, womit höchstens $2^{NI(X;Y)}$ Blöcke ausgewählt werden können, so dass die Ausgabeblöcke mit hoher Wahrscheinlichkeit unterscheidbar sind.

Für einen Kanal $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ ist die N-te Erweiterung der Kanal $(\mathcal{X}^N, \mathcal{Y}^N, P_{Y^N|X^N})$ mit

$$P_{Y^N|X^N} = \sum_{i=1}^{N} P_{Y|X}(y_i|x_i)$$

Ein (M,N)-Code für einen Kanal $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ ist eine Menge $\{\mathbf{x}(1), \ldots, \mathbf{x}(M)\} \subset \mathcal{X}^N$. Die Rate eines (M,N)-Codes ist:

$$R = \frac{\log M}{N}$$

Wir betrachten die Abbildung $\{1, \ldots, M\} \to \mathcal{X}^N, s \mapsto \mathbf{x}(s)$ als Codierer für Nachrichten $s \in \{1, \ldots, M\}$. Die $\mathbf{x}(s)$ heissen Codewörter, $\{\mathbf{x}(1), \ldots, \mathbf{x}(M)\}$

ist das Codebuch. Ein (M, N)-Code verwendet M von $|\mathcal{X}|^N$ Wörtern der Länge N als Codewörter. Ein Decodierer für einen (M, N)-Code ist eine Abbildung

$$\mathcal{Y}^N \to \{1, \ldots, M\}, \mathbf{y} \mapsto \hat{s}(\mathbf{y})$$

Für $s \in \{1, \ldots, M\}$ bezeichnen wir

$$\lambda_s = P(\hat{s}(Y^N) \neq s | X^N = \mathbf{x}(s))$$

als Blockfehlerwahrscheinlichkeit, gegeben dass s gesendet wurde. Die maximale Blockfehlerwahrscheinlichkeit ist $\lambda_{max} = \max_{s \in \{1,\ldots,M\}} \lambda_s$

Für einen gegebenen Kanal sagen wir, dass eine Rate erreichbar ist, falls es eine Folge von $(\lceil 2^{NR} \rceil, N)$-Codes mit zugehörigen Decodieren gibt, so dass gilt:

$$\lambda_{max}^{(N)} \xrightarrow{N \to \infty} 0$$

Gegeben sei ein Kanal $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$. Die Menge der gemeinsam typischen Sequenzen $(\mathbf{x}, \mathbf{y}) \in \mathcal{X}^N \times \mathcal{Y}^N$ für $\varepsilon > 0$ ist:

$$T_{N,\varepsilon}^{X,Y} = \left\{ (\mathbf{x}, \mathbf{y}) \in \mathcal{X}^N \times \mathcal{Y}^N | \left| -\frac{1}{N} \log P(\mathbf{x}) - H(X) \right| \leq \varepsilon \right.$$
$$\left| -\frac{1}{N} \log P(\mathbf{y}) - H(Y) \right| \leq \varepsilon$$
$$\left. \left| -\frac{1}{N} \log P(\mathbf{x}, \mathbf{y}) - H(X, Y) \right| \leq \varepsilon \right\}$$

Das gemeinsame AEP besagt: Seien (X^N, Y^N) Zufallsvariablen mit Werten in $\mathcal{X}^N \times \mathcal{Y}^N$ und mit Verteilung $P_{X^N|Y^N}(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^n P_{X,Y}(x_i, y_i)$ (die (X_i, Y_i) sind also unabhängig und identisch verteilt). Dann gilt:

1. $P((X^N, Y^N) \in T_{N,\varepsilon}^{X,Y}) \to 1$ für $N \to \infty$
2. $|T_{N,\varepsilon}^{X,Y}| \leq 2^{N(H(X,Y)+\varepsilon)}$
3. Falls $\tilde{X}^N$ und $\tilde{Y}^N$ unabhängige Zufallsvariablen mit $P_{\tilde{X}^N} = P_{X^N}$ und $P_{\tilde{Y}^N} = P_{Y^N}$ sind, dann gilt:

   $$P((\tilde{X}^N, \tilde{Y}^N) \in T_{N,\varepsilon}^{X,Y}) \leq 2^{-N(I(X;Y)-3\varepsilon)}$$

   Und für ausreichend grosses $N \gg 1$ gilt:

   $$P((\tilde{X}^N, \tilde{Y}^N) \in T_{N,\varepsilon}^{X,Y}) \geq (1-\varepsilon) 2^{-N(I(X;Y)+3\varepsilon)}$$

Beweis: **1.)** Gemäss dem schwachen Gesetz der grossen Zahlen existiert für gegebene ε, δ ein $N_0(X)$, so dass für $N > N_0(X)$

$$P\left(\left| -\frac{1}{N} \log P(X_1, \ldots, X_N) - H(X) \right| \geq \varepsilon \right) \leq \frac{\delta}{3}$$

Analog existieren $N_0(Y)$ und $N_0(X,Y)$ mit den entsprechenden Ungleichungen. Somit sind mit $N_0 := \max\{N_0(X), N_0(Y), N_0(X,Y)\}$ für $N \geq N_0$ mit Wahrscheinlichkeit $> 1-\delta$ alle drei Ungleichungen erfüllt, woraus folgt $P(T^{X,Y}_{N,\varepsilon}) > 1-\delta$

2.) Folgt aus $1 \geq \sum_{(\mathbf{x},\mathbf{y})\in T^{X,Y}_{N,\varepsilon}} P(\mathbf{x},\mathbf{y}) \geq |T^{X,Y}_{N,\varepsilon}| 2^{-N(H(X,Y)+\varepsilon)}$

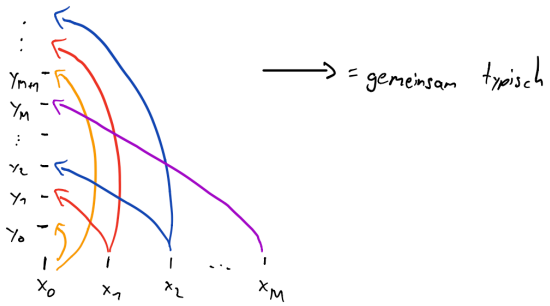
3.) Es gilt:

$$\begin{aligned} P\left(\left(\tilde{X}^N, \tilde{Y}^N\right) \in T^{X,Y}_{N,\varepsilon}\right) &= \sum_{T^{X,Y}_{N,\varepsilon}} P_{X^N}(\mathbf{x}) P_{Y^N}(\mathbf{y}) \\ &\leq 2^{N(H(X,Y)+\varepsilon)} 2^{-N(H(X)-\varepsilon)} 2^{-N(H(Y)-\varepsilon)} \\ &= 2^{-N(I(X;Y)-3\varepsilon)} \end{aligned}$$

Ausserdem gilt für ausreichend grosses N, dass $1-\varepsilon < P(T^{X,Y}_{N,\varepsilon})$ und damit $1-\varepsilon < \sum_{(\mathbf{x},\mathbf{y})\in T^{X,Y}_{N,\varepsilon}} P(\mathbf{x},\mathbf{y}) \leq |T^{X,Y}_{N,\varepsilon}| 2^{-N(H(X,Y)-\varepsilon)}$ und somit also $|T^{X,Y}_{N,\varepsilon}| \geq (1-\varepsilon)2^{N(H(H,Y)-\varepsilon)}$. Es folgt:

$$\begin{aligned} P\left(\left(\tilde{X}^N, \tilde{Y}^N\right) \in T_{N,\varepsilon}\right) &= \sum_{(\mathbf{x},\mathbf{y})\in T^{X,Y}_{N,\varepsilon}} P_{X^N}(\mathbf{x}) P_{Y^N}(\mathbf{y}) \\ &> (1-\varepsilon)2^{N(H(X,Y)-\varepsilon)} 2^{-N(H(X)+\varepsilon)} 2^{-N(H(Y)+\varepsilon)} \\ &= (1-\varepsilon)2^{-N(I(X;Y)+3\varepsilon)} \end{aligned}$$

Das gemeinsame AEP besagt also, dass für unabhängig gewählte Paare $(\mathbf{x},\mathbf{y}) \in T^X_{N,\varepsilon} \times T^Y_{N,\varepsilon}$ die Wahrscheinlichkeit, ein gemeinsam typisches $(\mathbf{x},\mathbf{y})$ zu erhalten, ca. $2^{-NI(X;Y)}$ beträgt (Teil 3). Somit können in $T^X_{N,\varepsilon}$ ca. $M = 2^{NI(X;Y)}$ Elemente $\mathbf{x}_i$ gefunden werden, so dass die Mengen $A_i := \{\mathbf{y}|\ (\mathbf{x},\mathbf{y}) \in T^{X,Y}_{N,\varepsilon}\}$ disjunkt sind (wobei dies nur eine Heuristik ist, Disjunktheit von A_i müsste gezeigt werden). Denn "nur" jedes $2^{NI(X;Y)}$-te Paar ist auch gemeinsam typisch (im Erwartungswert).



4.3 Shannons Kanalcodierungstheorem

1. Sei $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ ein Kanal mit Kapazität C. Dann ist jede Rate $R < C$ erreichbar, d.h. es existiert eine Folge von $(\lceil 2^{NR} \rceil, N)$-Codes und zugehörigen Decodieren, so dass $\lambda_{max}^{(N)} \xrightarrow{N\to\infty} 0$ gilt.
2. Umgekehrt gilt für jede Folge von $(\lceil 2^{NR} \rceil, N)$-Codes mit $\lambda_{max}^{(N)} \xrightarrow{N\to\infty} 0$, dass $R \leq C$.

Im Beweis wird für alle Codes der folgende Decodierer $\mathcal{Y}^N \to \{1, \ldots, \lceil 2^{NR} \rceil\}$ verwendet: Ein empfangenes $\mathbf{y} \in \mathcal{Y}^N$ wird als $\hat{s} \equiv \hat{s}(\mathbf{y})$ decodiert, wenn erfüllt ist:

- Das Paar $(\mathbf{x}(\hat{s}), \mathbf{y})$ ist gemeinsam typisch
- Es gibt kein $s' \neq \hat{s}$, so dass $(\mathbf{x}(s'), \mathbf{y})$ gemeinsam typisch ist.

Wobei wir davon ausgehen, dass ein bestimmtes $\varepsilon > 0$ und die entsprechende typische Menge $T_{N,\varepsilon}^{X,Y}$ fixiert wurden. Falls kein $\hat{s} \in \{1, \ldots, \lceil 2^{NR} \rceil\}$ beide Bedingungen erfüllt, werten wir dies als Decodierfehler.

Beweis:

4.3.1 Erreichbarkeit

1.) Sei P_X eine Verteilung über $\mathcal{X}$. Für gegebenes N, R sei $M := \lceil 2^{NR} \rceil$. Ein (M, N)-Code $\mathcal{C}$ ist gegeben durch Codewörter

$$\begin{aligned} \mathbf{x}(1) &= x_1(1)x_2(1)\ldots x_N(1) \\ &\vdots \\ \mathbf{x}(M) &= x_1(M)x_2(M)\ldots x_N(M) \end{aligned}$$

P_X erzeugt eine Verteilung auf der Menge aller $(\lceil 2^{NR} \rceil, N)$-Codes gemäss:

$$P(\mathcal{C}) = \prod_{s=1}^{M} \prod_{i=1}^{N} P_X(x_i(s))$$

Wir betrachten folgende Ereignisse:

- Ein zufälliger Code $\mathcal{C}$ wird gewählt und dem Sender / Empfänger bekannt gegeben.
- Eine Nachricht $s \in \{1, \ldots, M\}$ wird zufällig gemäss der Gleichverteilung $P_S(s) = \frac{1}{M}$ gewählt.
- Das Codewort $\mathbf{x}(s)$ wird über den Kanal gesendet.
- Der Empfänger decodiert das empfangene Wort $\mathbf{y} \in \mathcal{Y}^N$ via gemeinsamer typischer Decodierung wie oben beschrieben.

Nun wird die Wahrscheinlichkeit eines Decodierfehlers, also des Ereignisses $\mathcal{E} = \{\hat{s} \neq s\}$ analysiert (bzgl. der Wahl des Codes $\mathcal{C}$, der Nachricht s und des empfangenen Wortes $\mathbf{y}$). Die Wahrscheinlichkeit ist gegeben durch:

$$P(\mathcal{E}) = \sum_{\mathcal{C}} P(\mathcal{C}) \left[\frac{1}{M} \sum_{s=1}^{M} \lambda_s(\mathcal{C}) \right]$$

Wobei $\lambda_s(\mathcal{C})$ die Blockfehlerwahrscheinlichkeit bei gegebenem Code $\mathcal{C}$ und Nachricht s ist. Aus Symmetriegründen (wir summieren über alle Codes, wobei das Codewort für alle Indizes nach dem gleichen Schema erstellt wird), hängt $\sum_{\mathcal{C}} P(\mathcal{C})\lambda_s(\mathcal{C})$ nicht von s ab, somit:

$$\begin{aligned} P(\mathcal{E}) &= \frac{1}{M} \sum_{\mathcal{C}} \sum_{s=1}^{M} P(\mathcal{C})\lambda_s(\mathcal{C}) \\ &= \sum_{\mathcal{C}} P(\mathcal{C})\lambda_1(\mathcal{C}) \\ &= P(\mathcal{E}|s=1) \end{aligned}$$

Wir definieren für $i = 1, \ldots, M$:

$$E_i := \{\ (\mathbf{x}(i), \mathbf{y}) \in T_{N,\varepsilon}^{X,Y}\}$$

E_i beschreibt das Ereignis, dass beim Senden von $\mathbf{x}(1)$ für das empfangene $\mathbf{y} \in \mathcal{Y}^N$ das Paar $(\mathbf{x}(i), \mathbf{y})$ gemeinsam typisch ist. Somit wird das Eintreten eines Decodierfehlers beschrieben durch

$$E_1^C \cup E_2 \cup \ldots \cup E_M (= \mathcal{E} \cap \{s = 1\})$$

Wobei E_1^C das Ereignis "$(\mathbf{x}(1), \mathbf{y})$ sind nicht gemeinsam typisch" und $E_2 \cup \ldots E_M$ das Ereignis "es gibt ein $i \neq 1$, so dass $(\mathbf{x}(i), \mathbf{y})$ gemeinsam typisch sind" (wenn mindestens eines dieser beiden Ereignisse eintritt, passiert ein Decodierfehler). Mittels Union Bound ergibt sich:

$$\begin{aligned} P(\mathcal{E}|s=1) &= P(E_1^C \cup E_2 \cup \ldots \cup E_M) \\ &\leq P(E_1^C|s=1) + \sum_{i=2}^{M} P(E_i|s=1) \end{aligned}$$

Wegen der gemeinsamen AEP gilt $P(E_1^C|s=1) \xrightarrow{N\to\infty} 0$, d.h. für jedes $\varepsilon > 0$ gilt $P(E_1^C|s=1) \leq \varepsilon$ für ausreichend grosses N. $\mathbf{x}(1), \mathbf{x}(i), i \neq 1$ sind per Konstruktion unabhängig, somit auch $\mathbf{y}$ und $\mathbf{x}(i), i \neq 1$ (da die Verteilung von $\mathbf{y}$ durch $P(\mathbf{y}|\mathbf{x}(1))$ gegeben ist und aus der Unabhängigkeit folgt $P(\mathbf{y}|\mathbf{x}(1), \mathbf{x}(i)) = P(\mathbf{y}|\mathbf{x}(1))$). Die Wahrscheinlichkeit, dass $\mathbf{x}(i)$ und $\mathbf{y}$ gemeinsam typisch sind, ist somit (gemeinsames AEP Teil 3) $P(E_i|s=1) \leq$

$2^{-N(I(X;Y)-3\varepsilon)}$. Insgesamt folgt für grosses N und $R < I(X;Y) - 3\varepsilon$:

$$\begin{aligned} P(\mathcal{E}) &= P(\mathcal{E}|s=1) \\ &\leq P\left(E_1^c|s=1\right) + \sum_{s=2}^{M} P\left(E_i|s=1\right) \\ &\leq \varepsilon + (M-1)2^{-N(I(X;Y)-3\varepsilon)} \\ &\leq^{(*)} \varepsilon + 2^{-N(I(X;Y)-3\varepsilon-R)} \\ &\leq^{(**)} 2\varepsilon \end{aligned}$$

Wobei $(*)$ wegen $M - 1 \leq 2^{NR}$ folgt und $(**)$ für $R < I(X;Y) - 3\varepsilon$ gilt.

Insgesamt können wir also für gegebenes $R < I(X;Y)$ die Werte von ε und N so wählen, dass die durchschnittliche Fehlerwahrscheinlichkeit bzgl. möglicher $(\lceil 2^{NR} \rceil, N)$-Codes und Codewörter $\leq 2\varepsilon$ ist. Wählen wir für P_X nun eine Verteilung mit $I(X;Y) = C$ (die garantiert existiert, siehe Eigenschaften der Kanalkapazität), gilt die Folgerung mit C. Da die durchschnittliche Fehlerwahrscheinlichkeit $\leq 2\varepsilon$ ist, muss auch zwingend ein Code $\mathcal{C}^*$ mit $P(\mathcal{E}|\mathcal{C}^*) \leq 2\varepsilon$ existieren. Um dann aus $\mathcal{C}^*$ einen Code zu konstruieren, für den die maximale Blockfehlerwahrscheinlichkeit λ_{max} klein ist, werden die Hälfte der Codewörter entfernt. Da der Durchschnitt des Codes $\leq 2\varepsilon$ ist, müssen die $\frac{M}{2}$ "besten" Codewörter eine Blockfehlerwahrscheinlichkeit von $\leq 4\varepsilon$ haben (ansonsten wäre der Schnitt nicht erreichbar). Die Rate dieses Codes ist $\frac{1}{N} \log \frac{M}{2} = \frac{\log M}{N} - \frac{1}{N} \geq R - \frac{1}{N}$.

4.3.2 Kapazität als obere Schranke

Wenn wir einen fehlerfreien Code $(2^{NR}, N)$-Code $\mathcal{C}$ haben, dessen maximale Blockfehlerwahrscheinlichkeit verschwindet (d.h. $\lambda_{max}(\mathcal{C}) = 0$), dann ist die gesendete Nachricht s eindeutig durch das empfangene Y^N bestimmt und somit $H(S|Y^N) = 0$. Ist S gleichverteilt ($P(S = s) = \frac{1}{2^{NR}}$ für alle s), gilt $H(S) = NR$ und somit:

$$NR = H(S) = H\left(S|Y^N\right) + I\left(S;Y^N\right) \overset{(1)}{\leq} I\left(X^N;Y^N\right) \overset{(2)}{\leq} \sum_{i=1}^{N} I\left(X_i;Y_i\right) \overset{(3)}{\leq} NC$$

(1) gilt, weil $H(S|Y^N) = 0$ ist und $S \to X^N(S) \to Y^N$ eine Markov-Kette bilden (data processing inequality). (2) gilt wegen (wobei der dritte Schritt

aus der Gedächtnislosigkeit des Kanals folgt):

$$\begin{aligned} I\left(X^N;Y^N\right) &= H\left(Y^N\right) - H\left(Y^N|X^N\right) \\ &= H\left(Y^N\right) - \sum_{i=1}^{N} H\left(Y_i|Y_1,\ldots,Y_{i-1},X^N\right) \\ &= H\left(Y^N\right) - \sum_{i=1}^{N} H\left(Y_i|X_i\right) \\ &\leq \sum_{i=1}^{N} H\left(Y_i\right) - \sum_{i=1}^{N} H\left(Y_i|X_i\right) \\ &= \sum_{i=1}^{N} I\left(X_i;Y_i\right) \end{aligned}$$

Und (3) folgt aus der Definition der Kanalkapazität.

Im Allgemeinen ist die gesendete Nachricht S jedoch nicht eindeutig durch das empfangene Y^N bestimmt. Die Fano-Ungleichung besagt dann: Seien $S, \hat{S}$ Zufallsvariablen mit Wertebereich $\mathcal{S}$ und sei $P_e = P(\hat{S} \neq S)$. Dann gilt:

$$H(P_e, 1-P_e) + P_e \log|\mathcal{S}| \geq H(S|\hat{S})$$

Ist Y eine weitere Zufallsvariable, so dass $S \to Y \to \hat{S}$ eine Markowkette bildet, folgt daraus:

$$P_e \geq \frac{H(S|Y) - 1}{\log|\mathcal{S}|}$$

Beweis: Wir betrachten die Zufallsvariable

$$E = \begin{cases} 0, & \text{falls } \hat{S} = S \\ 1, & \text{falls } \hat{S} \neq S \end{cases}$$

Zwei Anwendungen der Kettenregel ergeben:

$$\begin{aligned} H(E,S|\hat{S}) &= H(E|S,\hat{S}) + H(S|\hat{S}) \\ &= H(S|E,\hat{S}) + H(E|\hat{S}) \end{aligned}$$

Und wir haben:

- $H(E|S,\hat{S}) = 0$, da E durch $S, \hat{S}$ bestimmt ist.
- $H(S|E,\hat{S}) = P(E=0)H(S|E=0,\hat{S}) + P(E=1)H(S|E=1,\hat{S}) \leq P_e \log|\mathcal{S}|$, weil $H(S|E=0,\hat{S}) = 0, P(E=1) = P_e$ und $H(S|E=1,\hat{S}) \leq \log|\mathcal{S}|$.
- $H(E|\hat{S}) \leq H(E) = H(P_e, 1-P_e)$

Womit zusammen mit obiger Gleichung die Aussage folgt. Die letzte Aussage folgt wegen $H(S|\hat{S}) \geq H(S|Y)$ (da $S \to Y \to \hat{S}$ eine Markovkette bilden wegen der data processing inequality) und $1 \geq H(P_e, 1 - P_e)$

Wir betrachten wieder die Situation, wo S gleichverteilt auf $\{1, \ldots, M\}$ ist und betrachten die durchschnittliche Fehlerwahrscheinlichkeit beim Versenden einer Nachricht $s \in \{1, \ldots, M\}$:

$$P_e = P(\hat{S} \neq S) = \frac{1}{M} \sum_s \lambda_s$$

Aus der Fano-Ungleichung folgt: Für einen Kanal mit Codebuch $\{\mathbf{x}(1), \ldots, \mathbf{x}(M)\} \subset \mathcal{X}^N$ und mit S gleichverteilt über $\{1, \ldots, M\}$ mit $M = \lceil 2^{NR} \rceil$ gilt:

$$H(S|\hat{S}) \leq 1 + P_e NR$$

Somit kann nun **2.)** vom Kanalcodierungstheorem bewiesen werden: Wenn für eine Folge von Codes die maximale Blockfehlerwahrscheinlichkeit $\lambda_{max}^{(N)} \to 0$ erfüllt, gilt auch $P_e^{(N)} \to 0$ (gemäss obiger Definition, d.h. die durchschnittliche Fehlerwahrscheinlichkeit für gleichverteiltes S), da $\lambda_{max} \geq P_e^{(N)} \geq 0$. Wir haben:

$$\begin{aligned} NR &\overset{(1)}{\leq} H(S) \\ &= H(S|\hat{S}) + I(S;\hat{S}) \\ &\overset{(2)}{\leq} 1 + P_e^{(N)} NR + I(S;\hat{S}) \\ &\overset{(3)}{\leq} 1 + P_e^{(N)} NR + I\left(X^N; Y^N\right) \\ &\overset{(4)}{\leq} 1 + P_e^{(N)} NR + NC \end{aligned}$$

(1) folgt, weil S gleichverteilt ist, (2) wegen der Fano-Ungleichung, (3) weil $S \to X^N \to Y^N \to \hat{S}$ eine Markovkette ist und (4) wegen $I(X^N; Y^N) \leq NC$, was wir bereits gezeigt haben. Also gilt

$$R \leq \frac{1}{N} + P_e^{(N)} R + C$$

und für $N \to \infty$ somit (wegen $P_e^{(N)} \to 0$) $R \leq C$.

4.4 Fehlerbehaftete Kommunikation oberhalb der Kapazität

Gegeben ist ein Kanal und ein zugehöriger (M, N)-Code. Wir repräsentieren Nachrichten $s \in \{1, \ldots, M\}$ als Wörter $\mathbf{s} = (s_1, \ldots, s_N) \in \{0,1\}^{\lceil \log M \rceil}$ und

betrachten für gegebenes $\mathbf{s}$ und $i = 1, \ldots, \lceil \log M \rceil$ die Wahrscheinlichkeit $P(\hat{s}_i \neq s_i)$ (d.h. die Wahrscheinlichkeit, dass sich die gesendete / decodierte Nachricht im i-ten Bit unterscheiden). Der Durchschnitt davon wird als durchschnittliche Bitfehlerwahrscheinlichkeit für diesen Kanal und Code bezeichnet:

$$p_b = \sum_{i=1}^{\lceil \log M \rceil} P(\hat{s}_i \neq s_i)$$

Wir nehmen an, dass wir einen rauschfreien Kanal mit Kapazität C haben (welcher aufbauend auf einem beliebigen Kanal immer konstruiert werden kann), d.h. $H(Y|X) = 0$. Wir können mit beliebigen Raten R kommunizieren, wenn wir die folgende Bitfehlerwahrscheinlichkeit in Kauf nehmen:

$$p_b = H_2^{-1}\left(1 - \frac{C}{R}\right)$$

Beweis: Wir betrachten für gegebenes $q \in [0, \frac{1}{2})$ einen (M, N)-Code für den BSC_q mit Rate $\frac{\log M}{N} \approx 1 - H_2(q)$ (Kapazität des BSC_q) und verschwindend kleiner Blockfehlerwahrscheinlichkeit (existiert wegen Shannons Theorem). Die Idee ist, den Decodierer dieses Codes zur Kompression von Binärwörtern der Länge N auf solche der Länge $\lceil \log M \rceil$ zu verwenden. Im Detail, zum Versand einer Nachricht $\{0,1\}^*$:

- Zerlege die Nachricht in Blöcke $\mathbf{x} \in \{0,1\}^N$
- Verwende den Decodierer g des BSC_q-Codes, um Blöcke $g(\mathbf{x}) \in \{0,1\}^{\lceil \log M \rceil}$ zu erhalten (verlustbehaftete Kompression)
- Sende $g(\mathbf{x})$ über den rauschfreien Kanal mit Kapazität C
- Verwende den Codierer des BSC_q-Codes, um aus $g(\mathbf{x})$ ein Wort $\hat{\mathbf{x}} \in \{0,1\}^N$ zu erhalten.

Die Kompressionsrate ist $\frac{N}{\lceil \log M \rceil} \approx \frac{1}{1-H_2(q)}$, womit die Kommunikation effektiv mit folgender Rate (die durch Wahl von q beliebig gross gemacht werden kann) erfolgt:

$$R \approx \frac{C}{1 - H_2(q)}$$

(auflösen nach q ergibt $q = p_b$ gemäss obiger Definition)

Für die Bitfehlerwahrscheinlichkeit (Anzahl Bits, in denen sich $\mathbf{x}$ und $\hat{\mathbf{x}}$ unterscheiden) betrachten wir ein beliebiges Codewort $f(\mathbf{s})$ des BSC_q-Codes und die Menge der $A_{f(\mathbf{s})} \subset \{0,1\}^N$ der Wörter, die typischerweise auftreten, wenn $f(\mathbf{s})$ über den Kanal gesendet wird (und deswegen von g auf $\mathbf{s}$ abgebildet werden). Dabei gilt:

- Elemente von $A_{f(\mathbf{s})}$ unterscheiden sich in ca. qN Bits von $f(\mathbf{s})$

- $A_{f(\mathbf{s})}$ hat Kardinalität $|A_{f(\mathbf{s})}| \approx 2^{NH(Y|X)} = 2^{NH_2(q)}$

Es gilt $M \approx 2^{N(1-H_2(q))}$ und somit $M|A_{f(\mathbf{s})}| \approx 2^{N(1-H_2(q))}2^{NH_2(q)} = 2^N = |\{0,1\}^N|$, womit fast alle Wörter in einer der M Mengen $A_{f(\mathbf{s})}$ liegen. Bei obigen Verfahren gibt es also typischerweise ein $\mathbf{s}$, so dass der zu sendende Block $\mathbf{x}$ in $A_{f(\mathbf{s})}$ liegt. Codierer / Decodier sind so gewählt, dass die Kommunikation fast fehlerfrei ist, womit typischerweise $g(\mathbf{x}) = \mathbf{s}$ und damit auch $\hat{\mathbf{x}} = f(g(\mathbf{x})) = f(\mathbf{s})$ gilt. $\mathbf{x}$ und $\hat{\mathbf{x}}$ unterscheiden sich also in ca. qN Bits und die Bitfehlerwahrscheinlichkeit ist ca. $p_b \approx q$ und somit:

$$p_b = H_2^{-1}\left(1 - \frac{C}{R}\right)$$

Kapitel 5

Elementare Codierungstheorie

5.1 Grundbegriffe

Die Hammingdistanz $d_H(\mathbf{x}, \mathbf{x}')$ zweier Wörter $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^N$ ist die Anzahl der Positionen, in denen sich $\mathbf{x}$ und $\mathbf{x}'$ unterscheiden. Die Minimaldistanz eines Codes $\mathcal{C}$ ist:

$$d_{\min}(\mathcal{C}) = \min_{\substack{x,x' \in \mathcal{C} \\ x \neq x'}} d_H\left(x, x'\right)$$

Für Kanäle mit identischem Ein- und Ausgabealphabet $\mathcal{X} = \mathcal{Y}$ ist eine naheliegende Strategie, ein empfangenes $\mathbf{y}$ als das nächste Codewort bzgl. der Hammingdistanz zu decodieren, d.h. $\hat{s} = \arg\min_s d_H(\mathbf{x}(s), \mathbf{y})$. Mit dieser Strategie können mindestens r Übertragungsfehler detektiert werden, wenn $d_{min}(\mathcal{C}) > r$ und mindestens s Übertragungsfehler korrigiert werden, wenn $d_{min}(\mathcal{C}) > 2s$. Somit kann das Worst-Case Verhalten anhand von $d_{min}(\mathcal{C})$ abgeschätzt werden.

5.2 Lineare Codes

Mit $\mathbb{F}_2$ wird die Menge $\{0, 1\}$ mit den folgenden Rechenoperationen bezeichnet:

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

$\mathbb{F}_2$ ist ein Körper und $\mathbb{F}_2^N = \mathbb{F}_2 \times \ldots \times \mathbb{F}_2$ ein $\mathbb{F}_2$-Vektorraum mit Operationen Addition ($(v, w) \mapsto v + w$ für $v, w \in \mathbb{F}_2^N$) und Skalarprodukt ($(\alpha, v) \mapsto \alpha \cdot v$ für $\alpha \in \mathbb{F}_2, v \in \mathbb{F}_2^N$). Ein Unterraum von $\mathbb{F}_2^N$ ist eine Teilmenge die selbst bezüglich dieser Operationen ein $\mathbb{F}_2$-Vektorraum ist (d.h. abgeschlossen bzgl. $+, \cdot$). Ein linearer $(2^K, N)$-Code über $\mathbb{F}_2$ ist ein K-dimensionaler

Unterraum von $\mathbb{F}_2^N$ und hat $M = 2^K$ Elemente. Jeder lineare Code enthält $\mathbf{0} = (0, \ldots, 0) \in \mathbb{F}_2^N$ (da es ein Unterraum ist). Für Codewörter $\mathbf{c} \in \mathcal{C}$ bezeichnet $w_H(\mathbf{c}) = d_H(\mathbf{c}, \mathbf{0})$ das Hamminggewicht von $\mathbf{c}$, d.h. die Anzahl der von 0 verschiedenen Einträge. Die Minimaldistanz von $\mathcal{C}$ kann damit geschrieben werden als:

$$d_{min}(\mathcal{C}) = \min_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}} w_H(\mathbf{c})$$

Denn $d_H(\mathbf{c}^1, \mathbf{c}^2) = d_H(\mathbf{c}^1 - \mathbf{c}^2, \mathbf{0}) = w_H(\mathbf{c}^1 - \mathbf{c}^2)$ für beliebige $\mathbf{c}^1, \mathbf{c}^2$ und $\mathbf{c}^1 - \mathbf{c}^2 \in \mathcal{C}$, da der Code linear ist.

5.3 Generatormatrizen

Jeder lineare $(2^K, N)$-Code $\mathcal{C}$ kann als Bild einer injektiven linearen Abbildung $\mathbb{F}_2^k \to \mathbb{F}_2^N$ betrachtet werden, d.h. es gibt eine $(K \times N)$-Matrix $\mathbf{G}$ mit Einträgen in $\mathbb{F}_2$, so dass jedes Codewort $\mathbf{c} \in \mathcal{C}$ die Form

$$\mathbf{c} = \mathbf{G}^T \cdot \mathbf{a}$$

hat für ein eindeutig bestimmtes $\mathbf{a} \in \mathbb{F}_2^K$. $\mathbf{a}$ und $\mathbf{c}$ werden dabei als Spaltenvektoren betrachtet und die Spalten von $\mathbf{G}^T$ bilden eine Basis von $\mathcal{C}$. Da $\mathcal{C}$ im Allgemeinen verschiedene (geordnete) Basen hat, ist $\mathbf{G}$ im Allgemeinen nicht eindeutig bestimmt. Ein spezifisches $\mathbf{G}$ entspricht einem spezifischen Codierer $\mathbb{F}_2^K \to \mathcal{C}$, der Nachrichten auf Codewörter abbildet.

Für $\mathcal{C} = \{000, 100, 010, 110\}$ sind bspw. mögliche Generatormatrizen:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Besonders geeignet sind Generatormatrizen der Form

$$\mathbf{G} = (\mathbf{I}_K, \mathbf{A})$$

wobei $\mathbf{I}_K$ die $K \times K$-Einheitsmatrix ist und $\mathbf{A}$ eine $K \times (N - K)$-Matrix. Solche $\mathbf{G}$ heissen systematisch. Wenn $\mathbf{G}$ systematisch ist, entsteht für ein $\mathbf{a} \in \mathbb{F}_2^K$ das Codewort $\mathbf{c} = \mathbf{G}^T \cdot \mathbf{a}$ durch Anhängen der Parity-Check-Bits $\mathbf{A}^T \cdot \mathbf{a} \in \mathbb{F}_2^{N-K}$ an die Informationsbits $\mathbf{a}$. Beispielsweise ist für $\mathcal{C} = \{0000, 0110, 1010, 1100\}$ die folgende Generatormatrix systematisch:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Die Parity-Check-Bits für $\mathbf{a} = (a_1, a_2) \in \mathbb{F}_2^2$ sind $(a_1 + a_2, 0) \in \mathbb{F}_2^2$

5.4 Parity-Check-Matrizen

Eine $(N-K) \times N$-Matrix $\mathbf{H}$ heisst Parity-Check-Matrix eines linearen Codes $\mathcal{C}$, falls

$$\mathbf{c} \in \mathcal{C} \Leftrightarrow \mathbf{H} \cdot \mathbf{c} = \mathbf{0}$$

Die Zeilen einer Parity-Check-Matrix $\mathbf{H}$ für einen Code $\mathcal{C}$ spannen also das orthogonale Komplement von $\mathcal{C}$ in $\mathbb{F}_2^N$ auf, d.h. den $(N-K)$-dimensionalen Unterraum

$$\mathcal{C}^\perp = \{\mathbf{v} \in \mathbb{F}_2^N | \langle \mathbf{v}, \mathbf{c} \rangle = 0 \text{ für alle } \mathbf{c} \in \mathcal{C}\}$$

Allgemein gilt für einen linearen Code mit systematischer Generatormatrix, dass $\mathbf{H} = (-\mathbf{A}^T, \mathbf{I}_{N-K})$ eine Parity-Check-Matrix für $\mathcal{C}$ ist. *Beweis:* Für $c \in \mathcal{C}$ existiert ein $a \in \mathbb{F}_2^K$ mit $\mathbf{G}^T \cdot \mathbf{a} = \mathbf{c}$ und somit $\mathbf{H} \cdot \mathbf{c} = \mathbf{H} \cdot \mathbf{G}^T \cdot \mathbf{a} = (-\mathbf{A}^T + \mathbf{A}^T) \cdot \mathbf{a} = \mathbf{0}$ Sei umgekehrt $\mathbf{c} \in \mathbb{F}_2^N$ ein Wort mit $\mathbf{H} \cdot \mathbf{c} = \mathbf{0}$. Schreiben wir $\mathbf{c} = (\mathbf{a}, \mathbf{b})$ mit $\mathbf{a} \in \mathbb{F}_2^K$ und $\mathbf{b} \in \mathbb{F}_2^{N-K}$, erhalten wir $\mathbf{H} \cdot \mathbf{c} = -\mathbf{A}^T \cdot \mathbf{a} + \mathbf{b} = \mathbf{0}$, d.h. $\mathbf{b} = \mathbf{A}^T \cdot \mathbf{a}$. Somit gilt also $\mathbf{G}^T \cdot \mathbf{a} = \mathbf{c}$, $\mathbf{c}$ ist also ein Codewort.

Für einen linearen Code $\mathcal{C}$ mit Parity-Check-Matrix $\mathbf{H}$ gilt:

$$d_{min}(\mathcal{C}) = \min\{l | \mathbf{H} \text{ hat } l \text{ linear abhängige Spalten } \}$$

Beweis: Die l Spalten $i_1, \ldots, i_l$ von $\mathbf{H}$ sind genau dann linear abhängig, wenn für $\mathbf{c} = (c_1 \ldots c_N) \in \mathbb{F}_2^N$ gegeben durch

$$c_j = \begin{cases} 1, & j \in \{i_1, \ldots, i_l\} \\ 0, & \text{sonst} \end{cases}$$

$\mathbf{H} \cdot \mathbf{c} = \mathbf{0}$ gilt, d.h. wenn $\mathbf{c} \in \mathcal{C}$ ist.

5.5 Hamming-Codes

Wir wählen ein $r > 1$, setzen $N := 2^r - 1$ und $K = N - r$ (und somit $r = N - K$) und betrachten eine $(N-K) \times N$-Matrix $\mathbf{H}$, deren Spalten die N von $\mathbf{0} \in \mathbb{F}_2^r$ verschiedenen Elemente von $\mathbb{F}_2^r$ sind. Durch Umordnen der Spalten können wir annehmen, dass $\mathbf{H}$ die folgende Gestalt für eine $K \times (N-K)$-Matrix $\mathbf{A}$ hat:

$$\mathbf{H} = (-\mathbf{A}^T, \mathbf{I}_{N-K})$$

Die $K \times N$-Matrix

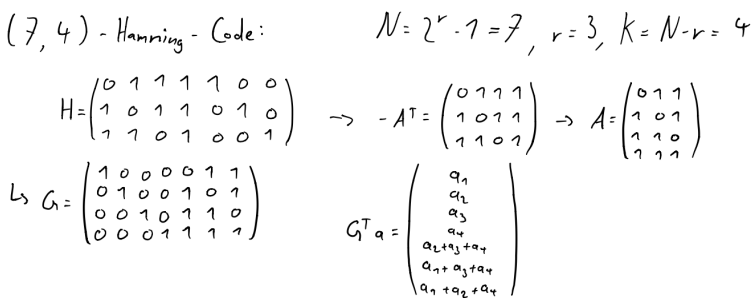
$$\mathbf{G} = (I_K, \mathbf{A})$$

definiert dann den (N, K)-Hamming-Code:

$$\mathcal{C} = \{ \mathbf{G}^T \cdot \mathbf{a} | \mathbf{a} \in \mathbb{F}_2^K\} \subset \mathbb{F}_2^N$$

Sämtliche Paare von Spalten von $\mathbf{H}$ sind per Konstruktion linear unabhängig, aber (viele) Tripel linear abhängig, womit $d_{min}(\mathcal{C}) = 3$ und ein Fehler pro Block korrigiert werden kann. Die Rate ist:

$$R = \frac{K}{N} = \frac{2^r - r - 1}{2^r - 1} = 1 - \frac{r}{2^r - 1} \xrightarrow{r\to\infty} 1$$



5.6 Syndrom-Decodierung linearer Codes

Sei $\mathcal{C}$ ein linearer Code mit Parity-Check-Matrix $\mathbf{H}$. Wenn $\mathbf{c} \in \mathcal{C}$ gesendet wird, wird im Allgemeinen

$$\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_2^N$$

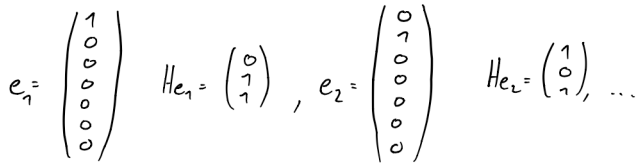
empfangen. Dabei ist σ das für den Fehlerterm $e \in \mathbb{F}_2^N$ charakteristische Syndrom:

$$\sigma = \mathbf{H} \cdot (\mathbf{c} + \mathbf{e}) = \mathbf{H} \cdot \mathbf{c} + \mathbf{H} \cdot \mathbf{e} = \mathbf{H} \cdot \mathbf{e}$$

Bei der Syndromdecodierung wird der Fehlerterm anhand des Syndroms σ geschätzt, z.B. anhand der Regel

$$\hat{e} = \underset{e\in\mathbb{F}_2^N, \mathbf{H}e=\sigma}{\arg\min}\, w_H(e)$$

welche das konsistente Fehlermuster mit dem geringsten Gewicht liefert. Es muss somit bei der Decodierung "nur" eine Tabelle von 2^{N-K} möglichen Syndromen betrachtet werden, was bei einem Code mit hoher Rate $\frac{K}{N}$ eine wesentliche Verbesserung sein kann. Für den $(7,4)$-Hamming-Code sieht dies bspw. folgendermassen aus:



5.7 Lineare Codes via Polynomevaluation

$\mathbb{F}_q$ ist ein endlicher Körper der Ordnung q (mit q Elementen). Er existiert genau dann, wenn $q = p^r$, wobei p prim und $r \in \mathbb{N}$ ist). Wir fassen Informationsvektoren $\mathbf{a} = (a_0, \ldots, a_{k-1}) \in \mathbb{F}_q^k$ als Polynome von Grad $\leq k-1$ über $\mathbb{F}_q$ auf:

$$p_{\mathbf{a}}(x) = a_0 + a_1 x^1 + \ldots + a_{k-1} x^{k-1}$$

Für gegebene $x_1, \ldots, x_N \in \mathbb{F}_q$ kann dann eine lineare Abbildung $\mathbb{F}_q^k \to \mathbb{F}_q^N$, deren Bild ein (q^k, N)-Code $\mathcal{C}$ ist, folgendermassen definiert werden:

$$\mathbf{a} \mapsto (p_{\mathbf{a}}(x_1), \ldots, p_{\mathbf{a}}(x_N))$$

Es gilt: Sei $\mathbb{F}$ ein Körper und seien $\alpha_0, \ldots, \alpha_d \in \mathbb{F}$ beliebige, paarweise verschiedene Stützstellen. Dann gibt es für jede Wahl von $\beta_0, \ldots, \beta_d \in \mathbb{F}$ ein eindeutig bestimmtes Polynom p vom Grad $\leq d$ mit $p(\alpha_i) = \beta_i$. *Beweis:* Existenz: Wähle

$$p(x) = \sum_{i=0}^{d} \beta_i l_i(x)$$

mit

$$l_i = \frac{(x-\alpha_0)\ldots(x-\alpha_{i-1})\,(x-\alpha_{i+1})\ldots(x-\alpha_d)}{(\alpha_i-\alpha_0)\ldots(\alpha_i-\alpha_{i-1})\,(\alpha_i-\alpha_{i+1})\ldots(\alpha_i-\alpha_d)}$$

($l_i(x) = 1$ für $x = \alpha_i$ und 0 für α_j mit $j \neq i$)

Eindeutigkeit: Sind p, p' zwei Polynomen mit dieser Eigenschaft, hat $p - p'$ (als Differenz von zwei Polynomen) Grad $\leq d$ und mindestens $d+1$ Nullstellen (nämlich $\alpha_0, \ldots, \alpha_d$), womit $p = p'$ folgt (da ein Polynom von Grad d höchstens d Nullstellen haben kann, ausser es ist konstant 0).

Daraus folgt, dass sich für $a \neq a' \in \mathbb{F}_q^K$ die Polynome $p_a, p_{a'}$ an mindestens $N-k+1$ der Stützstellen (Polynome sind von Grad $k-1$, bei $\geq k$ gleichen Stützstellen wären sie identisch) x_i unterscheiden müssen, somit gilt:

$$d_{min}(\mathcal{C}) \geq N - k + 1$$

Die Abbildung ist ausserdem injektiv (zwei verschiedene $a \neq a' \in \mathbb{F}_q^K$ werden auf verschiedene Elemente in $\mathbb{F}_q^N$ abgebildet) und damit tatsächlich ein (q^k, N)-Code.

In folgendem Beispiel ist die Codierung illustriert:

Gegeben: $q=4$, $k=3$, $q^k=64$ Wählen $N=4$ (somit $d_{min}(\mathcal{C}) \geq 2$), $x=(0,1,2,3)$

Übertragung von $a_1=(0,0,0)$: $p_{a_1}=0$, $a_1 \mapsto (0,0,0,0)$

$a_2=(0,1,1)$: $p_{a_2}=x+x^2$, $a_2 \mapsto (0,2,2,0)$

$\vdots$

5.7.1 Erreichbare Distanzen

Für beliebige N und $K \leq N$ und jedes $q \geq N$ ist die maximal erreichbare Minimaldistanz eines (q^K, N)-Codes über $\mathbb{F}_q$ gleich $N - K + 1$. *Beweis:* Eine Parity-Check-Matrix eines (q^K, N)-Codes $\mathcal{C}$ hat Format $(N - K) \times N$, also Spalten in $\mathbb{F}_q^{N-K}$. Dabei sind beliebige $N - K + 1$ Spalten linear abhängig, womit aus Abschnitt 5.4 $d_{min}(\mathcal{C}) \leq N - K + 1$ folgt. Im vorherigen Abschnitt haben wir einen Code mit $d_{min} \geq N - K + 1$ konstruiert, womit die Aussage folgt.

Davon ausgehend können wir lineare Codes über $\mathbb{F}_2$ mit grossem d_{min} konstruieren: Für jedes $r > 0$, $N \leq r2^r$ und $K \leq N - r$ gibt es einen linearen $(2^K, N)$-Code über $\mathbb{F}_2$ mit $d_{min} \geq n - k + 1$, wobei $n = \lfloor \frac{N}{r} \rfloor, k = \lceil \frac{K}{r} \rceil$. *Beweis:* Wir haben $k = \lceil \frac{K}{r} \rceil \leq \lceil \frac{N-r}{r} \rceil \leq \lfloor \frac{N}{r} \rfloor = n$ und $n = \lfloor \frac{N}{r} \rfloor \leq \lfloor \frac{r2^r}{r} \rfloor \leq 2^r$, insgesamt also $k \leq n \leq 2^r$. Somit existiert gemäss vorherigem Theorem (mit $q = 2^r$) ein linearer $(2^{rk}, n)$-Code über $\mathbb{F}_{2^r}$ mit $d_{min} = n - k + 1$, d.h. ein rk-dimensionaler Unterraum $\mathcal{C}_{2^r}$ von $\mathbb{F}_{2^r}^n$, gegeben als Bild einer linearen Abbildung $\mathbb{F}_{2^r}^k \to \mathbb{F}_{2^r}^n$. $\mathbb{F}_{2^r}$ ist ein $\mathbb{F}_2$-Vektorraum $\cong \mathbb{F}_2^r$ (jedes der 2^r Elemente kann entsprechend einem Vektor zugeordnet werden), weswegen die Abbildung als lineare Abbildung $\mathbb{F}_2^{rk} \to \mathbb{F}_2^{rn}$ aufgefasst werden kann. Dabei handelt es sich um einen linearen $(2^{rk}, rn)$-Code $\mathcal{C}_2$ über $\mathbb{F}_2$ mit $d_{min}(\mathcal{C}_2) \geq d_{min}(\mathcal{C}_{2^r}) = n - k + 1$ (Ungleichung folgt, da wenn sich zwei Stellen in $\mathbb{F}_{2^r}$ unterscheiden, dafür mindestens ein "Unterschied", potentiell jedoch sogar mehr in $\mathbb{F}_2^r$ generiert wird). Da $N \geq rn$ und $K \leq rk$, gibt es erst recht einen $(2^K, N)$-Code über $\mathbb{F}_2$ mit $d_{min} \geq n - k + 1$ (denn wenn wir weniger und längere Codewörter betrachten wird die Konstruktion nur einfacher).

5.8 Reed-Solomon-Codes

Sei $\alpha \in \mathbb{F}_q$ ein Element der Ordnung $N = q - 1$, d.h. $\alpha^N = 1$ und $\alpha^j \neq 1$ für jedes $1 < j < N$. Die zugehörige diskrete Fouriertransformation ist die lineare Abbildung

$$\mathcal{F}_\alpha : \mathbb{F}_q^N \to \mathbb{F}_q^N$$

die $\mathbf{v} = (v_0, \ldots, v_{N-1}) \in \mathbb{F}_q^N$ auf $\mathbf{w} = (w_0, \ldots, w_{N-1}) \in \mathbb{F}_q^N$ abbildet mit

$$w_i := \sum_{j=0}^{N-1} \alpha^{ij} v_j$$

Die inverse FT $\mathcal{F}_\alpha^{-1} : \mathbb{F}_q^N \to \mathbb{F}_q^N$ bildet $\mathbf{w} = (w_0, \ldots, w_{N-1})$ auf $\mathbf{v} = (v_0, \ldots, v_{N-1})$ ab mit:

$$v_i := -\sum_{j=0}^{N-1} \alpha^{-ij} w_j$$

Werden $\mathbf{v}, \mathbf{w}$ als Polynome $p_{\mathbf{v}}(x) = v_0 + v_1 x^1 + \ldots + v_{N-1} x^{N-1}$ interpretiert, gilt $w_i = p_{\mathbf{v}}(\alpha^i)$ und $v_i = -p_{\mathbf{w}}(\alpha^{-i})$.

Sei $\alpha \in \mathbb{F}_q$ ein Element der Ordnung $N = q - 1$ und sei $t < \frac{N}{2}$. Der zugehörige Reed-Solomon-Code ist der lineare (q^K, N)-Code mit $K = N - 2t$, dessen Codewörter alle $c \in \mathbb{F}_q^N$ sind, die erfüllen:

$$\mathcal{F}_\alpha(c) = (0, \ldots, 0, w_{2t}, \ldots, w_{N-1})$$

(d.h. die c, für welche die ersten $2t$ Einträge der Fourier-Transformierten 0 sind). Für die Codierung wird ein $\mathbf{w} \in \mathbb{F}_q^K$ als $(0, \ldots, 0, w_{2t}, \ldots, w_{N-1}) \in \mathbb{F}_q^N$ und $\mathcal{F}_\alpha^{-1}$ wird darauf angewandt. Bei der Decodierung wird $\mathcal{F}_\alpha$ auf das empfangene Wort $\mathbf{r} = \mathbf{c} + \mathbf{e}$ angewandt, womit man aufgrund der Linearität von $\mathcal{F}_\alpha$ erhält:

$$\underbrace{\mathcal{F}_\alpha(\mathbf{r})}_{=:R} = \underbrace{\mathcal{F}_\alpha(\mathbf{c})}_{=:C} + \underbrace{\mathcal{F}_\alpha(\mathbf{e})}_{=:E}$$

Die ersten $2t$ Koeffizienten von C sind gleich 0, somit kann $E_0, \ldots, E_{2t-1}$ abgelesen werden. Ist $w_H(E) \leq t$ (maximal t Fehler), können die restlichen Einträge von $E_{2t}, \ldots, E_N$ rekursiv bestimmt werden (siehe unten).

Die Minimaldistanz eines solchen RS-Codes ist $2t$. *Beweis:* Ein RS-Code ist per Definition gegeben durch:

$$\left\{ \mathcal{F}_\alpha^{-1}(0, \ldots, 0, w_{2t}, \ldots, w_{N-1}) \mid (w_{2t}, \ldots, w_{N-1}) \in \mathbb{F}_q^{N-2t} \right\}$$

Vektoren $(w_{2t}, \ldots, w_{N-1}) \in \mathbb{F}_q^{N-2t}$ können als Polynome aufgefasst werden:

$$\begin{aligned} p_w(x) &= w_{2t} x^{2t} + \cdots + w_{N-1} x^{N-1} \\ &= x^{2t} \left(w_{2t} + \cdots + w_{N-1} x^{N-1-2t} \right) \\ &= x^{2t} \tilde{p}_w(x) \end{aligned}$$

Die Einträge von c_i sind (per Definition der inversen Fouriertransformation) gegeben durch Evaluation von $-p_w$ an den Stellen $\alpha^{-i} \in \mathbb{F}_q, i = 0, \ldots, N-1$, d.h:

$$c_i = -\sum_{j=2t}^{N-1} w_j \alpha^{-ij} = -p_w\left(\alpha^i\right) = -\alpha^{2ti} \tilde{p}_w\left(\alpha^i\right)$$

Die Polynome $\tilde{p}_w$ sind von Grad $N - 2t - 1$, womit ihre Koeffizienten durch die Werte an $N - 2t$ Stellen eindeutig bestimmt sind. Da $p_w(x) = x^{2t} \tilde{p}_w$, sind auch die Polynome unter allen Polynomen dieser Form eindeutig durch $N - 2t$ Werte bestimmt (welche nicht Nullstellen von x^{2t} sind, was garantiert ist). Somit unterscheiden sich für verschiedene $w_0 \neq w_1$ die Polynome p_{w_0} und p_{w_1} an mindestens $2t + 1$ der Stellen $\alpha^i, i = 0, \ldots, N-1$, womit die Minimaldistanz $\geq 2t + 1$ ist. Die Minimaldistanz eines linearen (q^K, N)-Codes ist allgemein $\leq N - K + 1 = 2t + 1$, womit die Aussage folgt.

5.8.1 Fehlerkorrektur

Eine Folge $s = s_0, s_1, s_2, \ldots$ erfüllt eine lineare Rekursion der Länge L, falls es $c_1, \ldots, c_L$ gibt, so dass für alle $i \geq L$ gilt:

$$s_i = \sum_{j=1}^{L} c_j s_{i-j}$$

Fassen wir $s_0, s_1, s_2, \ldots$ als Potenzreihe

$$s(x) = s_0 + s_1 x + s_2 x^2 + \ldots$$

auf und die $c_1, \ldots, c_L$ als Polynom

$$f(x) = 1 - c_1 x - \ldots - c_L x^L$$

, so sind die Koeffizienten des Polynoms $s(x)f(x)$ für $i \geq L$ gegeben durch $s_i - \sum_{j=1}^{L} c_j s_{i-j} = 0$ und $s(x)f(x)$ damit ein Polynom von Grad $< L$. Umgekehrt gilt auch, dass wenn s eine Folge ist und f ein Polynom f mit der Form wie oben, so dass $s(x)f(x)$ ein Polynom von Grad $< L$ ist, dann erfüllt s die durch f definierte lineare Rekursion.

Ein Beispiel für eine lineare Rekursion der Länge 1 ist die Fouriertransformation $\mathbf{w} = (w_0, \ldots, w_{N-1}) \in \mathbb{F}_q^N$ eines Vektors $\mathbf{v} = (0, \ldots, v_k, 0, \ldots) \in \mathbb{F}_q^N$:

$$w_i = \sum_{j=0}^{N-1} \alpha^{ij} v_j = \alpha^{ik} v_k$$

Denn es gilt $w_{i+1} = \alpha^k w_i$.

Ausserdem gilt, dass für zwei lineare rekursive Folgen die Summe wieder eine linear rekursive Folge ist (wobei die Rekursionspolynome multipliziert werden).

Besitzt eine Folge $s_0, s_1, s_2, \ldots$ mindestens eine lineare Rekursion der Länge $\leq L$, dann kann eine solche aus $2L$ Werten $s_i, \ldots, s_{i+2L-1}$ durch Lösen eines linearen Gleichungssystems der Grösse $L \times L$ bestimmt werden. Denn wir erhalten für jedes $j = i + L, \ldots, i + 2L - 1$ eine lineare Gleichung $s_j = \sum_{k=1}^{L} c_k s_{j-k}$, somit also L Gleichungen für die L Unbekannten $c_1, \ldots, c_L$.

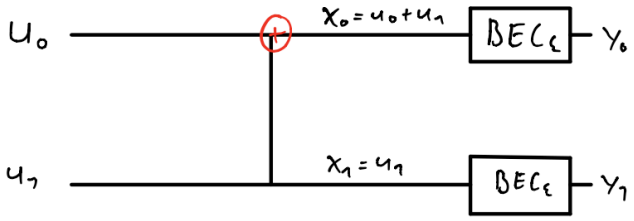
Ist $\mathbf{w} \in \mathbb{F}_q^N$ die FT eines Vektors $\mathbf{v} \in \mathbb{F}_q^N$ mit Gewicht t (t Einträge $\neq 0$), erfüllt $\mathbf{w}$ eine lineare Rekursion der Länge t (dies folgt aus dem Beispiel mit Gewicht 1 und der Eigenschaft, dass Summen ebenfalls eine lineare Rekursion mit Multiplikation der Polynome bilden). Da wir bei der Decodierung von RS-Codes die Einträge $E_0, \ldots, E_{2t-1}$ von $E = \mathcal{F}_\alpha(\mathbf{e})$ ablesen können, können wir für $\mathbf{e}$ mit Gewicht $\leq t$, das eine lineare Rekursion der Länge $\leq t$ hat, die Koeffizienten der Rekursion mit den bekannten Werten bestimmen, womit eine effiziente Korrektur von bis zu t Fehlern möglich ist.

5.9 Polar-Codes

Für jeden diskreten, gedächtnislosen, symmetrischen Kanal mit binärem Input und Kapazität C, jedes $R < C$ und jedes $\delta > 0$ gibt es einen lineare $(2^K, N = 2^n)$-Code mit Rate $> R$, durchschnittlicher Blockfehlerwahrscheinlichkeit $P_e < \delta$ und Codier- und Decodierkomplexität in $O(N \log N)$.

Polar-Codes werden nachfolgend für den BEC_ε illustriert. Wenn wir $N = 2$ Bits u_0, u_1 über zwei unabhängige Kopien des BEC_ε senden wollen, verwenden wir zur Codierung die lineare Abbildung

$$T_2 : \mathbb{F}_2^2 \to \mathbb{F}_2^2, \quad (u_0, u_1) \mapsto (x_0 = u_0 + u_1, x_1 = u_1)$$



Dabei betrachten wir die Ereignisse:

$$S_0 := \{u_0 \text{ kann anhand von } (y_0, y_1) \text{ bestimmt werden}\}$$
$$S_1 := \{u_1 \text{ kann anhand von } (y_0, y_1) \text{ bestimmt werden}\}$$

Wir haben für S_0 per Konstruktion $u_0 = x_0 + x_1$ und $u_1 = x_1$. Um u_0 zu bestimmen, muss x_0, x_1 bestimmt werden, was genau dann möglich ist, wenn $y_0 \neq ?$ und $y_1 \neq ?$. Somit:

$$P(S_0) = P(y_0 \neq ?)P(y_1 \neq ?) = 1 - (1 - \varepsilon)^2$$

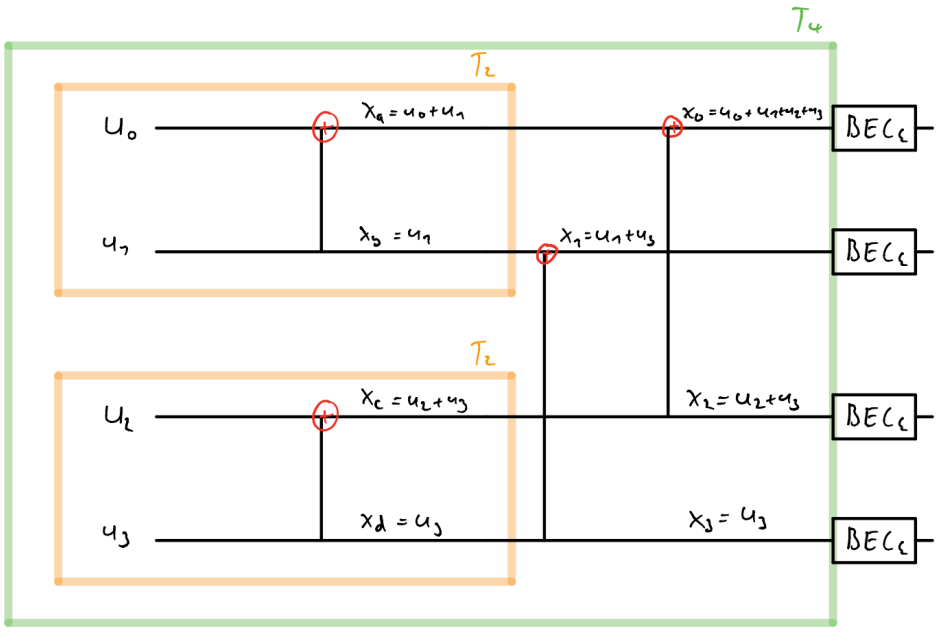
Und somit (wenn wir mit "$u_0 = ?$" das Ereignis bezeichnen, dass u_0 nicht bestimmt werden kann) $P(u_0 = ?|y_0, y_1) = (1 - \varepsilon)^2 = 2\varepsilon - \varepsilon^2$. Für S_1 gilt $u_1 = u_0 + x_0$ und $u_1 = x_1$. Wir nehmen nun an, dass u_0 bekannt ist, womit u_1 bestimmt werden kann, falls x_0 oder x_1 bekannt ist, d.h. falls $y_0 \neq ?$ oder $y_1 \neq ?$. Somit:

$$P(S_1) = 1 - P(y_0 = ?)P(y_1 = ?) = 1 - \varepsilon^2$$

bzw. $P(u_1 = ?|y_0, y_1, u_0) = \varepsilon^2$

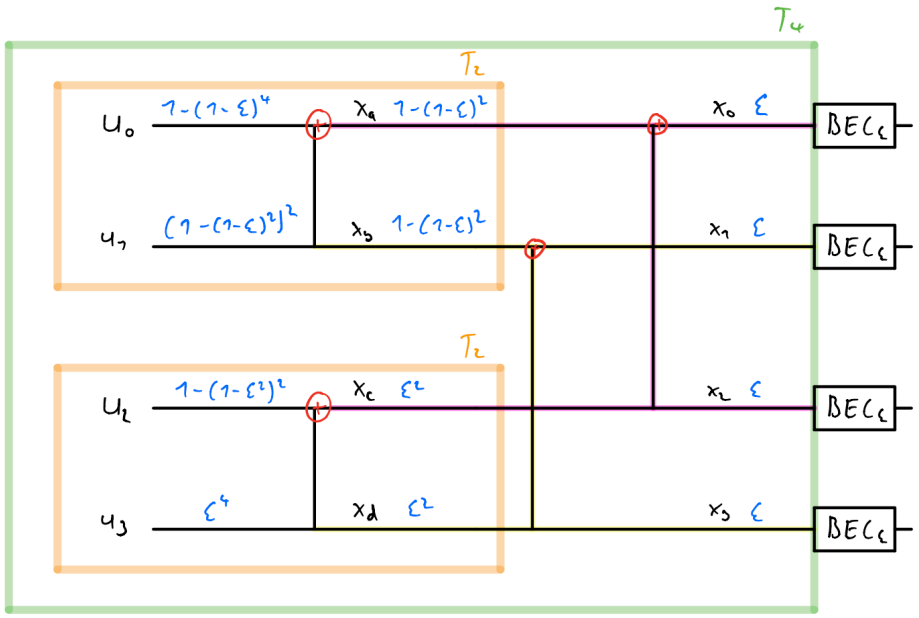
Für $N = 4$ wird die Konstruktion iteriert. Für die Codierung wird die lineare Abbilung $T_4 : \mathbb{F}_2^4 \to \mathbb{F}_2^4$ verwendet, gegeben durch die Matrix:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



Wir betrachten die Ereignisse

$$S_i := \{u_i \text{ kann anhand von } y_0, \ldots, y_3 \text{ und } u_0, \ldots, u_{i-1} \text{ bestimmt werden}\}$$



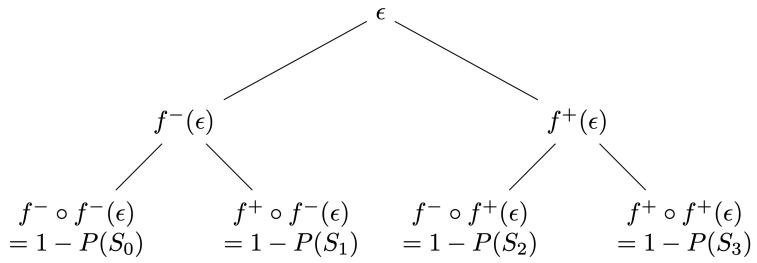
Es gilt $P(x_a = ?|y_0, y_1, y_2, y_3) = P(x_b = ?|y_0, y_1, y_2, y_3) = 1 - (1-\varepsilon)^2$, da es sich beim violetten / gelben Teil des Diagramms um eine Kopie von T_2 handelt (die Ereignisse sind ausserdem unabhängig, da $x_a = x_0 + x_2$ und $x_b = x_1 + x_3$). Weiter erhält man aus der Analyse von T_2:

$$\begin{aligned} P(u_0 = ?|y_0, y_1, y_2, y_3) &= 1 - \left(1 - \left(1 - (1-\varepsilon)^2\right)\right)^2 \\ &= \ldots = 1 - (1-\varepsilon)^4 \end{aligned}$$

Und $P(u_1 = ?|y_0, y_1, y_2, y_3) = (1 - (1-\varepsilon)^2)^2$. Wenn u_0, u_1 bekannt sind, sind es auch $x_a = u_0 + u_1$ und $x_b = u_1$. Somit folgt (da der violett / gelbe Teil eine Kopie von T_2 ist): $P(x_c = ?|y_0, y_1, y_2, y_3) = P(x_d = ?|y_0, y_1, y_2, y_3) = \varepsilon^2$. Schliesslich folgt:

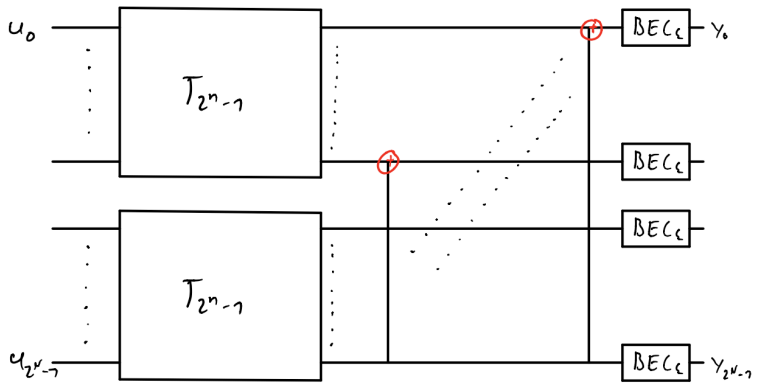
$$\begin{aligned} 1 - P(S_2) &= P(u_2 = ?|y_0, \ldots, y_3, u_0, u_1) = 1 - \left(1 - \varepsilon^2\right)^2 \\ 1 - P(S_3) &= P(u_3 = ?|y_0, \ldots, y_3, u_0, u_1, u_2) = \left(\varepsilon^2\right)^2 = \varepsilon^4 \end{aligned}$$

Die Auslöschwahrscheinlichkeiten entstehen dabei wie folgt:



Mit $f^-(x) := 1 - (1-x)^2$ und $f^+(x) := x^2$

Nach dem gleichen Vorgehen kann rekursiv eine lineare Abbildung $T_N : \mathbb{F}_2^N \to \mathbb{F}_2^N$ für alle $N = 2^n$ konstruiert werden:



Die Auslöschungswahrscheinlichkeiten

$$\varepsilon_{n,i} := 1 - P(S_i) = P(u_i = ?|y_0, \ldots, y_{2^n-1}, u_0, \ldots, u_{i-1})$$

sind die Werte $f^{i_1} \circ \ldots \circ f^{i_n}(\varepsilon)$, wobei $(i_1, \ldots, i_n) \in \{+, -\}^n$. Für $n \to \infty$ sind fast alle dieser Werte 0. Oder genauer: Betrachte für beliebiges $\delta > 0$ die Mengen

$$I_n^{<\delta} := \{i \in \{0, \ldots, 2^n - 1\} \,|\varepsilon_{n,i} < \delta\} \qquad \text{"gute Indizes"}$$
$$I_n^{>1-\delta} := \{i \in \{0, .., 2^n - 1\} \,|\varepsilon_{n,i} > 1 - \delta\} \qquad \text{"schlechte Indizes"}$$

Es gilt:

$$\lim_{n\to\infty} \frac{|I_n^{<\delta}|}{2^n} \to 1 - \varepsilon$$
$$\lim_{n\to\infty} \frac{|I_n^{>1-\delta}|}{2^n} \to \varepsilon$$

Der asymptotische Anteil der guten Indizes entspricht also genau der Kapazität des BEC_ε. Die Idee ist nun, Information in Bits n_i mit kleinem $\varepsilon_{n,i}$ zu packen. Zusammengefasst funktioniert die Codierung / Decodierung folgendermassen:

- Sei $R < 1 - \varepsilon$ und $N = 2^n, k = \lceil nR \rceil$ (wobei n so gross ist, dass $\frac{k}{n} < 1 - \varepsilon$)
- Berechne die $\varepsilon_{n,i}$ mit $i = 0, \ldots, N-1$ und setze

$$I := \{i_0, \ldots, i_{k-1} \in \{0, \ldots, 2^n - 1\} \text{ mit kleinstem } \varepsilon_{n,i_j}\}$$

- Codierung: Betrachte die lineare Abbildung $U_{N,k} : \mathbb{F}_2^k \to \mathbb{F}_2^N$ gegeben durch $(a_0, \ldots, a_{k-1}) \mapsto (u_0, \ldots, u_{N-1})$ mit

$$\begin{cases} u_{i_j} = a_j, & j \in \{0, \ldots, k-1\} \\ u_i = 0, & i \notin I \end{cases}$$

  d.h. Bits mit grossem $\varepsilon_{n,i}$ werden eingefroren. Sende dann

$$(x_0, \ldots, x_{N-1}) = T_N(u_0, \ldots, u_{N-1}) = (T_N \circ U_{N,k})(a_0, \ldots, a_{k-1})$$

  Wobei T_N die oben beschriebene Abbildung für $N = 2^n$ ist.

- Decodierung: Sei $(y_0, \ldots, y_{N-1})$ die Ausgabe des Kanals. Wir decodieren sukzessive $u_0, u_1, u_2, \ldots$ Angenommen, $u_0, u_1, \ldots, u_{i-1}$ sind bereits decodiert. Für u_i gilt:
  - Falls $i \notin I$: Wir wissen, dass $u_i = 0$ ist
  - Falls $i \in I$: Decodiere mit Hilfe der $y_0, \ldots, y_{N-1}$ und der $u_0, \ldots, u_{i-1}$ zunächst die benötigten Zwischenwerte x und dann u_i. Falls das nicht gelingt (weil eines der benötigten $y_i = ?$ ist, gib eine Fehlermeldung aus und stoppe die Decodierung.

Kapitel 6

Anhang

6.1 Jensen-Ungleichung

Für eine konvexe Funktion f und für nichtnegative λ_i mit $\sum_{i=1}^{n} \lambda_i = 1$ gilt:

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f\left(x_i\right)$$

Umgekehrt gilt für konkave Funktionen:

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \geq \sum_{i=1}^{n} \lambda_i f\left(x_i\right)$$

Gleichheit gilt genau dann für strikt konkave / konvexe f, wenn $x_1 = x_2 = \ldots = x_n$ oder f linear ist mit einer Domäne, die $x_1, x_2, \ldots, x_n$ enthält.

Die Ungleichung lässt sich insbesondere für Erwartungswerte anwenden, wenn f konvex ist gilt:

$$f(\mathrm{E}(X)) \leq \mathrm{E}(f(X)).$$