Information & Kommunikation - Zusammenfassung

Patrick Pletscher

29. September 2004

1. Grundlagen der Informationstheorie

1.1. Entropie als Mass für Unsicherheit

Definition der Entropie

Die Entropie einer *diskreten* Wahrscheinlichkeitsverteilung $[p_1, \ldots, p_L]$ ist

$$H([p_1, \ldots, p_L]) = -\sum_{i:1\leq i\leq L,\, p_i>0} p_i \log_2 p_i$$

Die Entropie der *Zufallsvariablen* X ist

$$-\sum_{x\in\mathcal{X}:P_X(x)\neq 0} P_X(x) \log_2 P_X(x)$$

falls diese Summe endlich ist.

binäre Entropiefunktion h

Die Entropie einer binären Zufallsvariablen X mit Parameter $P_X(0) = p$ wird als *binäre Entropiefunktion* h bezeichnet und ist durch

$$h(p) = -p \log_2 p - (1-p) \log_2 (1-p)$$

für $0 < p < 1$ und $h(0) = h(1) = 0$ definiert. Die Funktion $h(p)$ ist strikt konkav und besitzt ihr Maximum bei $p = 1/2$ mit $h(1/2) = 1$.

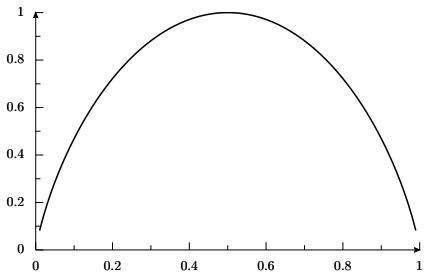

Abbildung 1: *Binäre Entropiefunktion*

Entropie als Erwartungswert

$H(x)$ kann auch als Erwartungswert einer reellwertigen Funktion $g(\cdot) = -\log_2 P_X(\cdot)$, ausgewertet für X, aufgefasst werden:

$$H(X) = E[g(X)] = E[-\log_2 P_X(X)] = E[\log_2 \frac{1}{P_X(x)}]$$

Schranken für die Entropie

Theorem 1 *Es gilt*

$$0 \leq H(X) \leq \log_2 |\mathcal{X}|$$

(oder äquivalent $0 \leq H([p_1, \ldots, p_L]) \leq \log_2 L$*) mit Gleichheit auf der linken Seite genau dann wenn* $P_X(x) = 1$ *für ein* x *und mit Gleichheit auf der rechten Seite genau dann, wenn alle Werte* $x \in \mathcal{X}$ *gleich wahrscheinlich sind.*

1.2. Entropiegrössen mehrerer Zufallsvariablen

Verbundentropie mehrerer Zufallsvariablen

Ein Paar $[X, Y]$, ein Tripel $[X, Y, Z]$ oder eine Liste $[X_1, \ldots, X_N]$ von Zufallsvariablen kann als eine einzelne, vektorwertige Zufallsvariable aufgefasst werden. Die gemeinsame Entropie mehrerer Zufallsvariablen ist also bereits definiert. Es gilt z.B.

$$H(XY) = -\sum_{(x,y)} P_{XY}(x,y) \log P_{XY}(x,y) = E[-\log P_{XY}(X,Y)]$$

Theorem 2 *Es gilt*

$$H(X) \leq H(XY)$$

Gleichheit gilt genau dann, wenn Y *durch Kenntnis von* X *eindeutig bestimmt ist*

Theorem 3 *Es gilt*

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

Gleichheit gilt genau dann, wenn X *und* Y *statistisch unabhängig sind.*

Bedingte Entropie und Information

Die *bedingte Entropie* von X, gegeben Y ist

$$H(X|Y) := H(XY) - H(Y)$$

und die *gegenseitige Information*, die X über Y gibt (und symmetrisch Y über X), ist

$$I(X;Y) := H(X) + H(Y) - H(XY)$$

$H(X|Y)$ ist die restliche Unsicherheit über X, wenn Y bekannt ist. Die Information $I(X;Y)$ ist die *Reduktion der Unsicherheit* über X, wenn man Y erfährt:

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

Theorem 4 *Es gilt*

$$0 \leq H(X|Y) \leq H(X)$$

Mit Gleichheit links genau dann, wenn X durch Y bestimmt ist. Die rechte Ungleichung ist äquivalent zu $I(X;Y) \geq 0$

Kettenregel

$$H(X_1X_2\dots X_n) = H(X_1\dots X_{n-1}) + H(X_n|X_1\dots X_{n-1})$$

Ein rekursives Muster für die Auflösung.

Bedingte Information

Die *bedingte gegenseitige Information*, die X über Y gibt, gegeben Z ist

$$I(X;Y|Z) := H(XZ) + H(YZ) - H(XYZ) - H(Z)$$

Es ist einfach zu sehen, dass $I(X;Y|Z) = H(X|Z) - H(X|YZ)$, d.h. $I(X;Y|Z)$ ist die Reduktion der Unsicherheit über X, wenn man Y erfährt, wobei Z schon bekannt ist.

Theorem 5 *Es gelten folgende Ungleichungen, die zudem äquivalent zueinander sind:*

$$I(X;Y|Z) \geq 0 \qquad \textit{und} \qquad H(X|YZ) \leq H(X|Z)$$

Die Aussage, dass Zusatzinformation die Entropie nicht vergrössern kann, gilt nicht für die Information: $I(X;Y|Z) > I(X;Y)$ ist möglich, nämlich, wenn $R(X;Y;Z)$ negativ ist.

Berechnung erhöht die Information nicht

Keine Berechnung kann die Information erhöhen, die die Daten zu irgendeiner Frage geben.

Die Zufallsvariablen X, Y, Z bilden eine *Markov-Kette*, geschrieben $X \to Y \to Z$, falls $P_{Z|XY}(z,x,y) = P_{Z|Y}(z,y)$ für alle x, y, z was äquivalent ist zu $H(Z|XY) = H(Z|Y)$ oder $I(Z;X|Y) = 0$.
Aus der Symmetrie $I(Z;X|Y) = I(X;Z|Y)$ folgt sofort auch die Umkehrbarkeit von Markov-Ketten: falls $X \to Y \to Z$ gilt auch $Z \to Y \to X$

Lemma 6 (Informationsverarbeitungs-Lemma) *Falls $X \to Y \to Z$, so gelten*

$$I(X;Z) \leq I(Y;Z) \qquad \textit{und} \qquad I(X;Z) \leq I(X;Y)$$

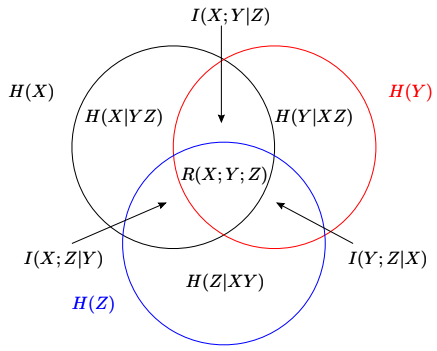

Abbildung 2: *Informationstheoretische Grössen für drei Zufallsvariablen*

Perfekt sichere Verschlüsselung

Ein Klartext M werde mit einem geheimen Schlüssel K zu einem Chiffrat C verschlüsselt. Der Empfänger kann mit Hilfe des Schlüssels das Chiffrat wieder entschlüsseln. Der Gegner sieht das Chiffrat, hat aber zu Beginn keine Information über den Schlüssel.

Ein solches Verfahren heisst *perfekt sicher* wenn $I(M;C) = 0$, d.h. wenn das Chiffrat statistisch unabhängig vom Klartext ist.

Theorem 7 *Jedes perfekt sichere Verschlüsselungssystem erfüllt $H(K) \geq H(M)$.*

2. Datenkompression

2.1. Codes für die Darstellung von Information

nicht-degeneriert

Ein *Code* C über dem Codealphabet $\mathcal{D}$ (mit $|\mathcal{D}| = D$) für eine Menge $\mathcal{X}$ ist eine Abbildung von $\mathcal{X}$ auf $\mathcal{D}^*$ (die Menge der Wörter über $\mathcal{D}$). Für $x \in \mathcal{X}$ bezeichnet $C(x)$ das Codewort für den Wert x und $l_C(x)$ die Länge von $C(x)$. Der Code C heisst *nicht-degeneriert*, wenn alle Codewörter verschieden sind, d.h. wenn aus $x_1 \neq x_2$ folgt, dass $C(x_1) \neq C(x_2)$, und wenn $C(x)$ für kein $x \in \mathcal{X}$ das leere Wort ist.

eindeutig decodierbar und präfixfrei

Ein Code C mit Codealphabet $\mathcal{D}$ heisst *eindeutig decodierbar*, wenn die Abbildung $\mathcal{X}^* \to \mathcal{D}^*$ definiert durch $[x_1 \parallel \dots \parallel x_n] \mapsto [C(x_1) \parallel \dots \parallel C(x_n)]$ eineindeutig ist. Der Code C heisst *präfixfrei*, wenn kein Codewort ein Präfix eines anderen Codewortes ist, d.h. wenn es keine zwei Codewörter c und c' gibt, so dass $c = c' \parallel d$ für irgendein $d \in \mathcal{D}^*$ mit $|d| \geq 1$.

Jeder präfixfreie Code ist eindeutig decodierbar, und jeder eindeutig decodierbare Code ist nicht-degeneriert.

optimaler Code

Ein Code C zur Codierung einer Zufallsvariable X ist *optimal*, wenn die mittlere Codewortlänge

$$E[l_C(X)] = \sum_{x \in \mathcal{X}} P_X(x) l_C(x)$$

minimal ist.

2.2. Codebäume und die Kraft'sche Ungleichung

Ein Code ist genau dann präfixfrei, wenn im entsprechenden Baum alle Codewörter Blätter sind. Wir nennen einen D-ären Baum *ausgefüllt*, wenn jeder innere Knoten genau D Nachfolgeknoten hat. Ein präfixfreier Code ist ausgefüllt, wenn der Codebaum ausgefüllt ist und jedem Blatt ein Codewort entspricht.

Sei $\mathcal{B}$ die Menge der Blätter und $P(b)$ für $b \in \mathcal{B}$ die Wahrscheinlichkeit des Blattes b, wobei $\sum_{b \in \mathcal{B}} P(b) = 1$ gilt. Die *Blattentropie* eines Baumes T ist definiert als

$$H_T = -\sum_{b \in \mathcal{B}} P(b) \log P(b)$$

Die *Tiefe* $t(b)$ eines Blattes b in einem Baum T ist seine Distanz von der Wurzel. Die mittlere Blatttiefe von T wird mit t_T bezeichnet und ist gegeben durch

$$t_T = \sum_{b \in \mathcal{B}} P(b) t(b)$$

Theorem 8 (Kraft'sche Ungleichung) *Ein D-ärer präfixfreier Code mit L Codewörtern der Längen $l_1, \ldots, l_L$ existiert genau dann wenn $\sum_{i=1}^{L} D^{-l_i} \leq 1$*

2.3. Schranken für die Codierung einer ZV

Theorem 9 *Die mittlere Codewortlänge $E[l_C(x)]$ eines optimalen präfixfreien Codes C über einem Codealphabet $\mathcal{D}$ mit $|\mathcal{D}| = D$ für eine ZV X erfüllt*

$$\frac{H(X)}{\log D} \leq E[l_C(X)] < \frac{H(X)}{\log D} + 1$$

Für den binären Fall ($D = 2$):

$$H(X) \leq E[l_C(X)] < H(X) + 1$$

2.4. Optimale Codes

Lemma 10 *Der Baum eines optimalen präfixfreien binären Codes ist ausgefüllt, d.h. er besitzt keine unbesetzten Blätter.*

Lemma 11 *Es gibt einen optimalen binären präfixfreien Code für X, in dem die beiden Codewörter für x_{L-1} und x_L sich nur im letzten Bit unterscheiden, d.h. in dessen Codebaum zwei Geschwisterblättern zugeordnet sind.*

Aus einem binären präfixfreien Code C für die Liste $[p_1, \ldots, p_L]$ von Codewort-Wahrscheinlichkeiten mit $p_1 \geq p_2 \geq \ldots p_L$ kann ein Code C' für die Liste $[p_1, \ldots, p_{L-2}, p_{L-1} + p_L]$ konstruiert werden, indem die beiden Blätter für p_{L-1} und p_L entfernt und ihr gemeinsamer Vorfahre als neues Blatt verwendet wird.

Lemma 12 *C ist ein optimaler binärer präfixfreier Code für die Liste $[p_1, \ldots, p_L]$ genau dann wenn C' optimal ist für die Liste $[p_1, \ldots, p_{L-2}, p_{L-1} + p_L]$.*

Theorem 13 (Huffman) *Der folgende Algorithmus liefert einen optimalen binären Code für die Liste $[p_1, \ldots, p_L]$ von Codewort-Wahrscheinlichkeiten.*

1. *Zeichne L Blätter (ohne Baum) mit Wahrscheinlichkeiten $p_1, \ldots, p_L$ und markiere sie als aktiv.*
2. *Führe den folgenden Schritt $(L-1)$-mal aus: Fasse zwei aktive Knoten mit minimalen Wahrscheinlichkeiten in einer Gabel zusammen, aktiviere neu die Wurzel der Gabel, weise ihr die Summe der beiden Wahrscheinlichkeiten zu, und deaktiviere die beiden zusammengefassten Knoten.*

Theorem 14 (McMillan) *Die Codewortlängen $l_1, \ldots, l_L$ jedes eindeutig decodierbaren Codes für ein Alphabet mit L Symbolen erfüllen ebenfalls die Kraft'sche Ungleichung $\sum_{i=1}^{L} D^{-l_i} \leq 1$. Insbesondere existiert immer ein präfixfreier Code mit den gleichen Codewortlängen.*

2.5. Nicht perfekte Datenkompression

Theorem 15 *Für die mittlere Bitfehlerwahrscheinlichkeit bei der Decodierung eines N-Bit-Strings X^N gilt*

$$\bar{P}_e \geq h^{-1}\left(\frac{H(X^N) - E[l_C(X^N)]}{N}\right)$$

Je grösser also die Differenz $H(X^N) - E[l_C(X^N)]$ zwischen Entropie und mittlerer Codewortlänge ist, desto grösser wird die mittlere Bitfehlerwahrscheinlichkeit.

2.6. Diskrete Informationsquelle

Definition von Informationsquellen

Eine *Informationsquelle* (oder ein *diskreter stochastischer Prozess*) ist eine unendliche Folge $\mathbf{X} = X_1, X_2, \ldots$ von ZV über einem Alphabet $\mathcal{X}$. Sie ist spezifiert durch die Liste der Wahrscheinlichkeitsverteilungen

$$P_{X_1}, P_{X_1 X_2}, P_{X_1 X_2 X_3}, \ldots$$

oder äquivalenterweise, durch die Liste der bedingten Wahrscheinlichkeitsverteilungen

$$P_{X_1}, P_{X_2|X_1}, P_{X_3|X_1 X_2}, \ldots$$

Eine Informationsquelle $\mathbf{X} = X_1, X_2, \ldots$ hat *endliches Gedächtnis* μ, falls jedes Symbol X_n höchstens von den μ

vorangehenden Symoblen $X_{n-\mu}, \ldots, X_{n-1}$, nicht aber weiter von der Vergangenheit abhängt, d.h. falls

$$P_{X_n|X_1 \ldots X_{n-1}} = P_{X_n|X_{n-\mu} \ldots X_{n-1}}$$

für $n = \mu + 1, \mu + 2, \ldots$. eine Quelle mit endlichem Gedächtnis $\mu = 1$ heisst *Markovquelle* und eine mit $\mu = 0$ heisst *gedächtnisfrei*.

Eine Informationsquelle $\mathbf{X} = X_1, X_2, \ldots$ heisst *stationär*, falls für jede Länge k eines Zeitfensters die Verteilung der k entsprechenden ZV nicht von der Position des Fensters abhängt, d.h. falls

$$P_{X_1 \ldots X_k} = P_{X_n \ldots X_{n+k-1}}$$

für alle $n = 1, 2, 3, \ldots$.

Entropierate von Informationsquellen

Die *Entropierate* $\bar{H}(\mathbf{X})$ einer Informationsquelle $\mathbf{X} = X_1, X_2, \ldots$ ist die mittlere Entropie pro Element der Folge

$$\bar{H}(\mathbf{X}) = \lim_{n \to \infty} \frac{1}{n} H(X_1 X_2 \ldots X_n)$$

wenn dieser Grenzwert existiert.

Gedächtnisfreie Quellen

Bei der *zeitinvarianten gedächtnisfreien Quelle* ist jedes Element X_n unabhängig von den vorherigen Elementen und alle Elemente die gleiche Verteilung besitzen: $P_{X_n} = P_X$ für alle n.

Theorem 16 *Für einen optimalen Code C für die Codierung von N unabhängigen Realisierungen $X_1, \ldots, X_N$ einer ZV X gilt*

$$\lim_{N \to \infty} \frac{E[l_C([X_1, \ldots, X_N])]}{N} = \frac{H(X)}{\log D}$$

Markovquellen und Quellen mit endlichem Gedächtnis

Eine Markovquelle $\mathbf{X} = X_1, X_2, \ldots$ heisst *zeitinvariant*, falls die bedingte Wahrscheinlichkeitsverteilung $P_{X_n|X_{n-1}}$ nicht von n abhängt.

Die *Zustandsübergangsmatrix* einer zeitinvarianten Markovquelle mit Alphabet $\mathcal{X} = \{v_1, \ldots, v_M\}$ ist die $M \times M$-Matrix $\mathbf{P} = [P_{ij}]$ mit

$$P_{ij} = P_{X_n|X_{n-1}}(v_i, v_j)$$

Die Spalten der Zustandsübergangsmatrix summieren jeweils zu eins, weil sie einer (bedingten) Wahrscheinlichkeitsverteilung entsprechen.

Eine zeitinvariante Markovquelle ist vollständig charakterisiert durch die Anfangsverteilunng P_{X_1} und die Zustandsübergangsmatrix. Falls man die Verteilung P_{X_n} als Vektor $(P_{X_n}(v_1), \ldots, P_{X_{n-1}}(v_m))^T$ auffasst, dann gilt

$$P_{X_n} = \mathbf{P} \cdot P_{X_{n-1}}$$

für alle n.

Eine Verteilung $\bar{P}_X$ über dem Alphabet einer Markovquelle heisst *stationäre Verteilung* $\bar{P}_X$, wenn sie zeitlich invariant bleibt, d.h. wenn

$$\bar{P}_X = \mathbf{P} \cdot \bar{P}_X$$

Eine Markovquelle ist stationär genau dann, wenn die Anfangsverteilung P_{X_1} stationär ist.

Zum Bestimmen der stationären Verteilung $\bar{P}_X$ verwendet man das Gleichungssystem $\bar{P}_X = \mathbf{P} \cdot \bar{P}_X$ zusammen mit der Zusatzgleichung, dass sich die Werte von $\bar{P}_X$ zu 1 summieren:

$$\begin{bmatrix} & \mathbf{P} & \\ 1 & 1 & 1 \end{bmatrix} \cdot \bar{P}_X = \begin{bmatrix} \bar{P}_X \\ 1 \end{bmatrix}, \text{ wobei } \bar{P}_X = \begin{bmatrix} \bar{P}_X(v_1) \\ \vdots \\ \bar{P}_X(v_M) \end{bmatrix}$$

Das Gleichungssystem enthält $M + 1$ Gleichungen für M Unbekannte, man kann eine beliebige ausser der letzten weglassen.

Die meisten Markovquellen haben nur eine stationäre Verteilung. Eine solche Markovquelle nennt man *ergodisch*. Ergodisch bedeutet, dass die Quelle nur ein stationäres Verhalten hat. Zu dieser strebt sie asymptotisch aus jeder Anfangsverteilung. Eine hinreichende (aber nicht notwendige) Bedingung für Ergodizität ist, dass alle Einträge der Zusandsübergangsmatrix echt positiv sind.

Der grösste Eigenwert der Zustandsübergangsmatrix $\mathbf{P}$ einer ergodischen Markovquelle ist immer 1, und der zugehörige Eigenvektor ist die stationäre Verteilung. Eine beliebige Anfangsverteilung der Zustände konvergiert exponentiell schnell zur stationären Verteilung, und der zweitgrösste Eigenwert von $\mathbf{P}$ ist der Exponent, der die Konvergenzgeschwindigkeit charakterisiert.

Im folgenden Theorem ist

$$H(X_n|X_{n-1} = v_j) = H([P_{1j}, \ldots, P_{Mj}])$$

die *Verzweigungsentropie* im Zustand i des Zustandsübergangsdiagramms.

Theorem 17 *Die Entropierate einer zeitinvarianten ergodischen Markovquelle mit Zustandsübergangsmatrix $\mathbf{P}$ und stationärer Verteilung $\bar{P}_X$ ist gegeben durch die gewichtete Verzweigungsentropie*

$$\bar{H}(\mathbf{X}) = -\sum_{ij} \bar{P}_X(v_j) P_{ij} \log P_{ij} = \sum_v \bar{P}_X(v) H(X_n|X_{n-1} = v)$$

Man multipliziert also die W'keit in einem Zustand zu sein mit der Entropie welche über den nächsten Zustand herrscht, danach summiert man dies über alle Zustände.

2.7. Universelle Datenkompression

Ein Datenkompressionsverfahren heisst *universell* für eine gegebene Klasse von Informationsquellen, wenn es, asymptotisch betrachtet, bei immer grösserer Wahl der Parameter

(z.B. Blocklänge), jede Quelle dieser Klasse auf die Entropierate komprimiert.

Präfixfreie Codes für die ganzen Zahlen

Für $j \in \mathbb{N}^+$ bezeichnen wir die normale binäre Codierung von j mit $B(j)$ und deren Länge mit

$$L(j) = \lfloor \log j \rfloor + 1$$

Der Code B ist nicht präfixfrei. Um einen ersten präfixfreien Code C_1 zu erhalten, können wir dem Codewort $B(j)$ eine Folge von $L(j) - 1$ Symbolen "0" voranstellen:

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

Die Länge des Codewortes $C_1(j)$ ist also

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

Dieser Code kann signifikant verbessert werden, wenn wir die $L(j) - 1$ Symbole "0" durch eine präfixfreie Codierung von $L(j)$ ersetzen. Als präfixfreie Codierung können wir z.B. C_1 verwenden. Weil jedes Codewort in B immer mit einer "1" beginnt, kann diese weggelassen werden. Den resultierenden Code bezeichnen wir mit B' und erhalten als präfixfreie Codierung der positiven ganzen Zahlen:

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

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

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

Beispiel: Codierung von $j = 37$. $B(37) = 100101$, $L(37) = 6$, $C_1(37) = 00000100101$, $C_1(L(37)) = C_1(6) = 00110$, $L_1(37) = 11$ (dezimal), $C_2(37) = 0011000101$ und $L_2(37) = 10$ (dezimal).

Intervalllängen-Codierung

Eine sehr effiziente Codierung für stationäre binäre Quellen $\mathbf{X} = X_1, X_2, \ldots$ mit starker Asymmetrie ist die sogenannte Intervalllängen-Codierung. Es sei $P_{X_i}(0) = 1 - p$ und $P_{X_i}(1) = p$ für alle $i \geq 1$ und für $p \ll 1/2$. Eine lange Folge solcher ZV wird codiert, indem nur die Abstände zwischen aufeinanderfolgenden Symbolen "1" codiert werden, und zwar mit einem geeigneten präfixfreien Code für die positiven ganzen Zahlen (hier C_2).

Die mittlere Anzahl Bits pro ZV ist also gegeben durch

$$\frac{E[L_2(D)]}{E[D]} \leq 2p \log(1 - \log p) - p \log p + p$$

Die Effizienz der Codierung wird für $p \to 0$ optimal:

$$\lim_{p \to 0} \frac{E[L_2(D)]}{E[D]h(p)} = 1$$

Wir betrachten nun die Verallgemeinerung für eine Q-äre stationäre Quelle (nicht notwendigerweise gedächtnisfrei). Das Alphabet sei $\mathcal{A} = \{a_1, \ldots, a_Q\}$, die Quelle sei $\mathbf{Y} = Y_1, Y_2, \ldots$ mit $P_{Y_1} = P_{Y_2} = \ldots = P_Y$. Der Unterschied zur binären Codierung ist, dass für *jedes* $y \in \mathcal{A}$ die Intervalllänge bis zum *letzten Auftreten des gleichen Symbols aus* $\mathcal{A}$ codiert werden muss. Für die Initialisierung nehmen wir an, dass $Y_{-i} = a_{i+1}$ für $i = 0, \ldots, Q - 1$.

3. Kommunikation über fehlerbehaftete Kanäle

3.1. Einleitung

Behandlung von Bitübertragungsfehlern

Bei einem *binäre symmetrische Kanal (BSK)*, welcher viele praktische Übertragungskanäle gut modelliert, wird jedes gesendete Bit mit einer bestimmten W'keit ϵ (Bitfehler-W'keit genannt) invertiert und mit W'keit $1 - \epsilon$ korrekt übertragen. Die auftretenden Fehler sind unabhängig voneinander.

Shannons Antwort

Für jeden Kanal lässt sich die sogenannte Kapazität angeben; dies ist die maximale Rate, mit der Information beliebig zuverlässig übertragen werden kann.

3.2. Diskrete gedächtnisfreie Kanäle

Grundsätzlich ist ein Kanal durch die bedingte W'keitsverteilung der Outputfolge, gegeben die Inputfolge, charakterisiert. Wenn wir die Symbolfolgen am Kanalinput und -output mit $X_1, \ldots, X_N$ und $Y_1, \ldots, Y_N$ bezeichnen, so wird der Kanal durch $P_{Y_1 \ldots Y_N | X_1 \ldots X_N}$ charakterisiert.

diskreter gedächtnisfreier Kanal (DGK)

Ein *diskreter gedächtnisfreier Kanal (DGK)* für ein Inputalphabet $\mathcal{X}$ und ein Outputalphabet $\mathcal{Y}$ ist eine bedingte W'keitsverteilung

$$P_{Y|X} : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}^+$$

es wird also jedes Inputsymbol unabhängig von den früheren Inputsymbolen behandelt.

3.3. Codierung und Decodierung

Blockcodes

Ein *Blockcode* C mit Blocklänge N für einen Kanal mit Inputalphabet $\mathcal{X}$ ist eine Teilmenge $C = \{\mathbf{c_1}, \ldots, \mathbf{c_M}\} \subseteq \mathcal{X}^N$ der N-Tupel über dem Alphabet $\mathcal{X}$, wobei $\mathbf{c_1}, \ldots, \mathbf{c_M}$ Codewörter genannt werden. Die *Rate* R von C ist definiert als

$$R = \frac{\log_2 M}{N}$$

d.h. R ist gleich der Anzahl Bits, die pro Kanalbenutzung gesendet werden können.

Optimale Schätzungen

Es soll eine ZV U auf Grund einer Beobachtung V geschätzt werden. Die bedingte W'keitsverteilung $P_{V|U}$, d.h. wie die Wirkung V von der Ursache U abhängt, ist bekannt. Ein *Schätzverfahren* ist eine Funktion f, welche jedem Wert v der Beobachtung den entsprechenden Schätzwert $\hat{u} = f(v)$ zuweist. Die ZV, die dem Schätzwert entspricht ist $\hat{U}$.

Eine Schätzung ist optimal, wenn die W'keit eines Fehlers $P(U \neq \hat{U})$, minimal ist. Es gilt

$$P(U = \hat{U}) = \sum_v P(U = \hat{U}, V = v)$$

wobei $\hat{U} = f(V)$ ist. Die Wahrscheinlichkeit in der Summe lässt sich anders schreiben:

$$P(U = \hat{U}) = \sum_v P_{UV}(f(v), v) = \sum_v P_{V|U}(v, f(v))P_U(f(v))$$

Dieser Ausdruck soll nun durch die Wahl der Schätzungsfunktion f maximiert werden. Dies wird erreicht, indem für jeden Wert v der Ausdruck in der Summe maximiert wird. Dabei muss man zwei Fälle unterscheiden:

1. P_U ist bekannt. In diesem Falll muss man für jedes v derjenige Wert $\hat{u}$ für $f(v)$ gewählt werden, für welchen

   $$P_{V|U}(v, \hat{u})P_U(\hat{u})$$

   maximal ist. Diese Schätzregel wird oft *Minimum-Error*-Regel (ME-Regel) genannt.

2. P_U ist nicht bekannt. Nimmt man an, alle Werte von U seien gleich wahrscheinlich, d.h. $P_U(u)$ sei für alle u gleich und müsse deshalb bei der Maximierung nicht beachtet werden, so erhalten wir folgende Regel. Für jedes v wird derjenige Wert $\hat{u}$ für $f(v)$ gewählt, der

   $$P_{V|U}(v, \hat{u})$$

   maximiert. Diese Schätzregel wird *Maximum-Likelihood*-Regel (ML-Regel) genannt.

ME- und ML-Decodierung

Wenn das Codewort

$$\mathbf{c_j} = [c_{j1}, \ldots, c_{jN}]$$

über einen DGK mit Übergangsverteilung $P_{Y|X}$ gesendet wird, so ist der Kanaloutput eine ZV $\mathbf{Y^N} = [Y_1, \ldots, Y_N]$ mit Wertemenge $\mathcal{Y}^N$ und W'keitsverteilung

$$P_{\mathbf{Y^N}|\mathbf{X^N}}(\mathbf{y^N}, \mathbf{c_j}) = \prod_{i=1}^{N} P_{Y|X}(y_i, c_{ji})$$

Im Decoder stellt sich das Problem, für ein empfangenes Kanaloutputwort

$$\mathbf{y^N} = [y_1, \ldots, y_N]$$

die beste Schätzung für das gesendete Codewort zu finden. Dieser Vorgang heisst *Decodierung* und entspricht dem im vorherigen Abschnitt besprochenen Schätzproblem, wobei $U = \mathbf{X^N}$, $V = \mathbf{Y^N}$, und $\hat{U}$ die Entscheidung des Decoders ist.

Theorem 18 (Minimum-Error Decoder) *Ein Decoder, der für ein gegebenes Empfangswort $\mathbf{y^N}$ als Schätzung des gesendeten Codewortes eines derjenigen $\mathbf{c_j} \in C$ wählt, welches*

$$P_{\mathbf{Y^N}|\mathbf{X^N}}(\mathbf{y^N}, \mathbf{c_j}) \cdot P_{\mathbf{X^N}}(\mathbf{c_j})$$

maximiert, erreicht die minimale mittlere Fehlerwahrscheinlichkeit.

Theorem 19 (Maximum-Likelihood Decoder) *Ein Decoder, der für ein gegebenes Empfangswort $\mathbf{y^N}$ eines derjenigen $\mathbf{c_j} \in C$ als Schätzung des gesendeten Codewortes wählt, welches*

$$P_{\mathbf{Y^N}|\mathbf{X^N}}(\mathbf{y^N}, \mathbf{c_j})$$

maximiert, erreicht die minimale mittlere Fehlerwahrscheinlichkeit, wenn alle Codewörter gleich wahrscheinlich sind.

3.4. Kanalkapazität und das Kanalcodierungstheorem

Definition

Die *Kapazität* eines durch $P_{Y|X}$ charakterisierten DGK ist das Maximum über Inputverteilungen P_X von $I(X;Y)$:

$$C = \max_{P_X} I(X;Y) = \max_{P_X} [H(Y) - H(Y|X)]$$

Falls folgende zwei Bedingungen erfüllt sind, ist die Kapazität einfach zu berechnen.

(A) $H(Y|X = x)$ ist für alle x gleich: $H(Y|X = x) = t$ für alle x.

(B) $\sum_x P_{Y|X}(y, x)$ ist für alle y gleich.

Theorem 20 *Die Kapazität eines Kanals mit Bedingung (A) ist $(\max_{P_X} H(Y)) - H(Y|X = x)$ (für irgendein x) und die Kapazität eines Kanals mit Bedingungungen (A) und (B) ist*

$$C = \log |Y| - t$$

Der BSK erfüllt beide Bedingungen und hat Kapazität $C = 1 - h(\epsilon)$. Der binäre Auslöschungskanal erfüllt Bedingung (A) und hat Kapazität $C = 1 - \delta$

Kapazität: Obergrenze für die Rate

Information kann nicht mit einer Rate über der Kanalkapazität zuverlässig übertragen werden.

Wir betrachten die Übertragung von K Informationsbits $U_1, \ldots, U_K$ durch N Benutzungen eines DGK. Die *Übertragungsrate* R ist demnach

$$R = \frac{K}{N} \qquad \text{(bits pro Kanalbenutzung)}$$

Es gilt

$$\begin{aligned} H(U^K|\hat{U}^K) &= H(U^K) - I(U^K; \hat{U}^K) \\ &\geq H(U^K) - NC \end{aligned}$$

d.h. die Kanalübertragung kann (mit oder ohne Feedback) die Unsicherheit über $U_1, \ldots, U_K$ beim Empfänger nicht um mehr als NC reduzieren. Damit U^K durch $\hat{U}^K$ eindeutig bestimmt ist ($H(U^K|\hat{U}^K) = 0$), muss die Anzahl N der Kanalbenutzungen mindestens $H(U^K)/C$ sein.

Theorem 21 (Shannon) *Wenn ein DGK mit Kapazität C zur Übertragung echt zufälliger Informationsbits mit Rate $R > C$ benutzt wird, so gilt für die mittlere Bitfehler-W'keit beim Empfänger*

$$h(\bar{P}_e) \geq 1 - \frac{C}{R}$$

Die Kapazität ist erreichbar

$R < C$ ist sowohl eine *notwendige* als auch *hinreichende* Bedingung für zuverlässige Übertragung.

Theorem 22 (Shannon) *Gegeben sei ein DGK mit Inputalphabet $\mathcal{X}$, Outputalphabet $\mathcal{Y}$ und Kapazität C. Für jede Rate $R < C$ und für jedes $\epsilon > 0$ existiert für genügend grosse Blocklänge N ein Code C mit $M = \lceil 2^{RN} \rceil$ Codewörtern, für den die maximale Decodierfehler-Wahrscheinlichkeit (über alle Codewörter) kleiner als ϵ ist, d.h.*

$$\max_{1 \leq j \leq M} P(\mathcal{F}|X^N = \mathbf{c_j}) \leq \epsilon$$

4. Elemente der Codierungstheorie

4.1. Minimaldistanz

Wir betrachten Codes mit $M = q^K$ Codewörtern für eine ganze Zahl $K < N$ (N = Blocklänge) und für $|\mathcal{X}| = q$ (hier meist binäre Codes $q = 2$).

Mit einem Code werden K q-äre Zeichen in einem Codewort der Länge N codiert. Das Verhältnis K/N der Anzahl Informationssymbole und der Anzahl Codesymbole wird oft als *dimensionslose Rate* bezeichnet.

Die *Hammingdistanz* $d(\mathbf{a}, \mathbf{b})$ zweier Wörter $\mathbf{a}$ und $\mathbf{b}$ (gleicher Länge über dem gleichen Alphabet) ist die Anzahl Positionen, in denen sich $\mathbf{a}$ und $\mathbf{b}$ unterscheiden. Die *Minimaldistanz* $d_{min}(\mathcal{C})$ eines Codes $\mathcal{C}$ ist die kleinste Hammingdistanz zwischen zwei Codewörtern.

r Fehler zu *detektieren* heisst das für *jedes* Codewort und *jedes* Fehlermuster mit höchstens r Fehlern festgestellt werden kann, dass ein Fehler aufgetreten ist, d.h. dass die Minimaldistanz des Codes grösser als r ist.
s Fehler zu *korrigieren* heisst wenn für jedes Codewort und für jedes Fehlermuster mit höchstens s Fehlern das Codewort wieder eindeutig gefunden werden kann. Dies ist genau dann der Fall, wenn die $d_{min} \geq 2s + 1$.

Theorem 23 *Ein Code mit Minimaldistanz d erlaubt, $d - 1$ Fehler zu detektieren oder $\lfloor (d-1)/2 \rfloor$ Fehler zu korrigieren.*

4.2. Lineare Codes

Ein *linearer Blockcode* mit q^K Codewörtern der Blocklänge N über einem endlichen Körper $GF(q)$ ist ein Unterraum der Dimension K des Vektorraumes der N-Tupel über $GF(q)$. Ein solcher Code wird als $[N, K]$-Code bezeichnet.

Jeder lineare Code enthält das Nullwort $\mathbf{0}$. Die Distanz eines Codewortes $\mathbf{c}$ zu $\mathbf{0}$, d.h. die Anzahl der von 0 verschiedenen Kompontenten, wird als *Hamminggewicht* $w(\mathbf{c})$ bezeichnet.

Theorem 24 *Die Minimaldistanz eines linearen Codes ist gleich dem minimalen Hamminggewicht eines von $\mathbf{0}$ verschiedenen Codewortes.*

Die Generatormatrix eines linearen Codes

Die Codierung eines *Informationsvektors* $\mathbf{a} = [a_1, \ldots, a_K]$ zu einem Codewort $\mathbf{c} = [c_1, \ldots, c_N]$ kann als Multiplikation mit einer $(K \times N)$-Matrix $\mathbf{G}$, der sogenannten *Generatormatrix*, betrachtet werden:

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

Die Zeilen der Generatormatrix bilden eine Basis des Codes.

Es gibt eine speziell geeignete Form einer Generatormatrix, nämlich wenn die ersten K Spalten der $(K \times K)$-Einheitsmatrix $\mathbf{I_K}$ entsprechen:

$$\mathbf{G} = [\mathbf{I_K} \vdots \mathbf{A}]$$

wobei $\mathbf{A}$ eine beliebige $K \times (N - K)$-Matrix ist. Dies bedeutet, dass das Codwort aus den K Informationssymbolen und $N - K$ angefügten "Parity-Checks" besteht. Eine solche Generatormatrix wird als *systematisch* bezeichnet.

Die Parity-Check-Matrix eines linearen Codes

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

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

Mit anderen Worten: Die Zeilen von $\mathbf{H}$ spannen den $(N-K)$-dimensionalen Unterraum aller Vektoren (N-Tupel) $\mathbf{v}$ auf, für die das Produkt $\mathbf{c} \cdot \mathbf{v^T}$ mit einem beliebigen Codewort $\mathbf{c}$ aus $\mathcal{C}$ null ergibt.

Theorem 25 *Sei $\mathbf{I_t}$ die $(t \times t)$-Einheitsmatrix und $\mathbf{G} = [\mathbf{I_K} \vdots \mathbf{A}]$ eine systematische Generatormatrix eines linearen Codes. Die $((N-K) \times N)$-Matrix $\mathbf{H} = [-\mathbf{A^T} \vdots \mathbf{I_{N-K}}]$ ist eine Parity-Check-Matrix des Codes.*

Theorem 26 *Die Minimaldistanz eines linearen Codes mit Parity-Check-Matrix $\mathbf{H}$ ist gleich der minimalen Anzahl Spalten in $\mathbf{H}$, die linear abhängig sind.*

Syndromdecodierung linearer Codes

Ein gesendetes Codewort $\mathbf{c}$ werde durch ein Fehlervektor $\mathbf{e}$ verfälscht, so resultiert aus der Multiplikation mit $\mathbf{H^T}$ ein für das Fehlermuster charakteristisches Bitmuster, das sogenannte *Syndrom* $\mathbf{s}$, welches unabhängig vom Codewort ist.

$$\mathbf{s} = (\mathbf{c} + \mathbf{e}) \cdot \mathbf{H^T} = \mathbf{c} \cdot \mathbf{H^T} + \mathbf{e} \cdot \mathbf{H^T} = \mathbf{0} + \mathbf{e} \cdot \mathbf{H^T} = \mathbf{e} \cdot \mathbf{H^T}$$

Man kann $\mathbf{e}$ auf Grund des Syndroms $\mathbf{s} = \mathbf{e} \cdot \mathbf{H^T}$ schätzen (ML) und danach vom empfangenen Wort subtrahieren.

Hamming-Codes und duale Hamming-Codes

Eine einfache Klasse von binären linearen Codes sind die sogenannten *Hamming-Codes*. Für jedes $r > 1$ gibt es ein Code der Länge $N = 2^r - 1$ mit $K = N - r$ Informationsbits und Minimaldistanz 3. Jeder Hamming-Code kann einen Fehler pro Block korrigieren.

Die $(r \times (2^r - 1))$-Parity-Check-Matrix eines Hamming-Codes kann sehr einfach konstruiert werden. Sie besteht aus allen $2^r - 1$ vom Nullvektor verschiedenen Spaltenvektoren.

Ein Code $\mathcal{C}'$ heisst *dual* zu einem Code $\mathcal{C}$, wenn die Generatormatrix von $\mathcal{C}'$ eine Parity-Check-Matrix von $\mathcal{C}$ ist.

Theorem 27 *In einem dualen Hamming-Code mit Parameter r gilt*

$$d_{min} = 2^{r-1}$$

4.3. Codes basierend auf Polynomevaluation

Theorem 28 (Singleton Bound) *Die Minimaldistanz eines linearen $[N, K]$-Codes über $GF(q)$ ist höchstens $N-K+1$.*

Lemma 29 *Jedes Polynom vom Grad d über einem beliebigen Körper lässt sich eindeutig aus den Polynomwerten an beliebigen $d+1$ Stützstellen interpolieren.*

Falls das Polynom an $\alpha_0, \ldots, \alpha_d$ ausgewertet wurde, so kann $f(x)$ wie folgt bestimmt werden:

$$f(x) = \sum_{i=0}^{d} \beta_i L_i(x)$$

mit

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

Theorem 30 *Für beliebige N und $K \leq N$ und jeden Körper $GF(q)$ mit $q \geq N$ ist die maximal erreichbare Minimaldistanz eines linearen $[N, K]$-Codes gleich $N - K + 1$.*

Dies können wir tun indem wir einen $[N, K]$-Code über $GF(q)$ konstruieren, indem wir die K Informationssymbole als die Koeffizienten eines Polynoms vom Grad $K-1$ auffassen. Das zu diesem Informationsvektor gehörende Codewort erhält man durch Auswertung des Polynoms an N fixen und bekannten Stellen $x_1, \ldots, x_N$.

Korollar 31 *Für jedes $r > 0$, $N \leq r2^r$ und $K \leq N-r$ gibt es einen linearen binären $[N, K]$-Code mit Minimaldistanz mindestens $n - k + 1$, wobei $n = \lfloor N/r \rfloor$ und $k = \lceil K/r \rceil$.*

4.4. Codes basierend auf Polynommultiplikation

Einem Polynom

$$a_d x^d + a_{d-1} x^{d-1} + \ldots + a_1 x + a_0$$

vom Grad d entspricht ein Vektor $[a_0, \ldots, a_d]$ mit $d+1$ Komponenten. Dieses Polynom kann aber natürlich auch als Vektor mit $N > d+1$ Komponenten aufgefasst werden, wobei die ersten $N - d - 1$ Komponenten null sind:

$$[a_0, \ldots, a_d, 0, \ldots, 0]$$

Ein $[N, K]$-Polynomcode über dem Körper $GF(q)$ ist durch das *Generatorpolynom*

$$g(x) = g_{N-K} x^{N-K} + g_{N-K-1} x^{N-K-1} + \ldots + g_1 x + g_0$$

vom Grad $N - K$ über $GF(q)$ wie folgt definiert. Die Informationsvektoren sind die q^K Polynome $i(x)$ über $GF(q)$ vom Grad $\leq K-1$ und das zu $i(x)$ gehörende Codewort ist das Polynom $i(x)g(x)$.

Die folgende Matrix ist also eine Generatormatrix dieses Codes:

$$\begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{N-K} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{N-K-1} & g_{N-K} & 0 & \cdots & 0 \\ 0 & 0 & g_0 & \cdots & g_{N-K-2} & g_{N-K-1} & g_{N-K} & \cdots & 0 \\ \vdots & \vdots & & \ddots & & \ddots & \ddots & \ddots & \\ \vdots & \vdots & & & & \ddots & \ddots & \ddots & \cdots \\ 0 & 0 & 0 & \cdots & g_0 & \cdots & g_{N-K-2} & g_{N-K-1} & g_{N-K} \end{bmatrix}$$

Codierung und Fehlerdetektion

Die Codierung entspricht einer Multiplikation des Informationspolynoms mit dem Generatorpolynom.

Das empfangene Wort (resp. Polynom) hat also die Form

$$r(x) = i(x)g(x) + e(x)$$

Während die Fehlerkorrektur für solche Codes nur in speziellen Fällen effizient möglich ist, ist eine Fehlerdetektion wie bei allen linearen Codes durch eine Prüfung des Syndroms sehr einfach möglich.
Wenn keine Fehler aufgetreten sind, so kann der Empfänger die Information $i(x)$ decodieren, indem er das empfangene Polynom $r(x)$ wieder durch $g(x)$ dividiert. Der Rest der Division (ein Polynom vom Grad höchstens $N-K-1$) ist gleich

$$e(x) \mod g(x)$$

und ist unabhängig von $i(x)$. $e(x)$ ist gleich dem 0-Polynom, falls keine Fehler aufgetreten sind, oder wenn $e(x)$ ein Vielfaches von $g(x)$ ist.
Oft treten Fehler aber gehäuft in sogenannten "Bursts" auf. Wenn alle auftretenden Fehler auf ein Fenster der Länge l konzentriert sind, so sprechen wir von einem Fehlerburst der Länge l.

Theorem 32 *Der durch das Generatorpolynom (vom Grad $N-K$) $g(x)$ generierte Code kann einen beliebigen Fehlerburst der Länge höchstens $N-K$ detektieren (sofern nur ein solcher Burst auftritt).*

Fehlerdetektierende CRC-Codes

In der Praxis verwendete und standardisierte Polynome sind

$$\begin{aligned} \text{CRC-12} &= x^{12} + x^{11} + x^3 + x^2 + x + 1 \\ \text{CRC-16} &= x^{16} + x^{15} + x^3 + x^2 + 1 \\ \text{CRC-CCITT} &= x^{16} + x^{12} + x^5 + 1 \end{aligned}$$

4.5. Reed-Solomon-Codes

Für Reed-Solomon-Codes gibt es eine effiziente Fehlerkorrekturprozedur. RS-Codes sind lineare Codes über einem Körper der Form $GF(q)$. In den meisten Anwendungen ist $q = 2^d$, d.h. die Körperelemente können als d-Bit-Strings dargestellt werden.

Definition der Reed-Solomon-Codes

Ein Reed-Solomon-Code über $GF(q)$ mit Parameter t und Element α der Ordnung $N = q - 1$ ist der lineare $[N, K]$-Code mit $N = q-1$ und $K = N-2t$, der aus allen Codworten $[c_0, \ldots, c_{N-1}]$ besteht, für welche die Fourier-Transformierte (für α) die Form

$$[\underbrace{0, \ldots, 0}_{2t}, C_{2t}, \ldots, C_{N-1}]$$

besitzt. Mit anderen Worten, ein Fenster der Länge $2t$ im Frequenzbereich ist auf Null gesetzt.

Die Minimaldistanz eines solchen Codes ist gleich $2t + 1$ und es gibt auch einen effizienten Algorithmus um bis zu t Fehler zu korrigieren.
Ein Polynom $c(x)$ ist genau dann ein Codewortpolynom, wenn

$$c(1) = c(\alpha) = c(\alpha^2) = \ldots = c(\alpha^{2t-1}) = 0$$

was äquivalent ist zur Aussage, dass das Polynom

$$g(x) = (x-1)(x-\alpha)(x-\alpha^2)\ldots(x-\alpha^{2t-2})(x-\alpha^{2t-1})$$

das Polynom $c(x)$ teilt. Somit ist ein RS-Code ein Polynomcode mit Generatorpolynom $g(x)$.

Effiziente Fehlerkorrektur

Eine mögliche Art der Codierung besteht darin, die $N - 2t$ Informationssymbole direkt den Werten $C_{2t}, \ldots, C_{N-1}$ der Fourier-Transformierten des Codewortes zuzuweisen und das Codewort mittels inverser Fourier-Transformation zu berechnen. Sei also

$$r(x) = c(x) + e(x)$$

das empfangene Wort, wobei $e(x)$ das Fehlermuster ist, dessen Gewicht höchstens t sei. Man kann die Fourier-Transformierte von $r(x)$ berechnen, wobei wegen der Linearität

$$R(x) = C(x) + E(x)$$

gilt. Da die untersten $2t$ Koeffizienten von $C(x)$ gleich 0 sind, kennt man also die untersten $2t$ Koeffizienten $E_0, \ldots, E_{2t-1}$ von $E(x)$. Diese Folge erfüllt eine lineare Rekursion der Länge gleich der Anzahl Fehler, also wegen unserer Annahme höchstens t ist. Die Rekursionsgleichung der Länge t kann man aus den $2t$ aufeinanderfolgenden Werten $E_0, \ldots, E_{2t-1}$ bestimmen. Anschliessend kann man die restlichen Koeffizienten $E_{2t}, \ldots, E_{N-1}$ von $E(x)$ gemäss der gefundenen linearen Rekursion berechnen. Nun muss nur noch $E(x)$ von $R(x)$ subtrahieren und erhält $C(x)$, worin die gesendete Information enthalten ist.

4.6. Fehlerbündel und Interleaving

Oft treten Fehler nicht unabhängig voneinander auf, sondern treten gehäuft auf; dies bezeichnet man als *Fehlerbündel*.
Im Folgenden werden wir das sogenannte *Interleaving* betrachten. Dies ist eine Methode, um gehäuft auftretende Fehler über mehrere Codewörter zu verteilen, so dass innerhalb eines einzelnen Codewortes nur noch wenige Fehler vorkommen, welche dann korrigiert werden können.

Definition

Wir bezeichnen einen Fehler als *Fehlerbündel* oder *Fehlerburst* der Länge b, falls alle Fehler in einem String innerhalb eines Intervalls der Länge b liegen.
Das *Interleaving* macht aus einem Code $\mathcal{C}$ durch Konkatenation mehrerer Codewörter und anschliessendem Umgruppieren der einzelnen Symbole einen Code $\mathcal{C}'$ mit verbesserten Korrektureigenschaften, so dass zum Beispiel längere Fehlerbursts korrigiert werden können.
Beim *Interleaving zur Tiefe* t wird aus einem linearen $[N, K]$-Code $\mathcal{C}$, der alle Fehlerbündel bis zur Länge b korrigiert, ein $[t\cdot N, t\cdot K]$-Code $\mathcal{C}'$ generiert, der alle Fehlerbündel bis zur Länge $t\cdot b$ korrigiert. Dazu wird dieser Code $\mathcal{C}'$ definiert als die Menge aller Codewörter der Form

$$c' = c_1^{(1)} c_1^{(2)} \ldots c_1^{(t)} c_2^{(1)} c_2^{(2)} \ldots c_2^{(t)} \ldots c_N^{(1)} c_N^{(2)} \ldots c_N^{(t)}$$

wobei $c^{(i)} = c_1^{(i)} c_2^{(i)} \ldots c_N^{(i)}$ (für $i = 1, \ldots, t$) Codewörter aus $\mathcal{C}$ sind. (Tiefgestellt bezeichnet Position im Codewort, Hochgestellt bezeichnet das i-te Codewort).

A. Fourier-Transformation

$$\mathbf{V} = \mathbf{F}\cdot\mathbf{v}$$

mit

$$\mathbf{F} = \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \ldots & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \ldots & \alpha^{1(N-1)} \\ \alpha^0 & \alpha^2 & \alpha^4 & \ldots & \alpha^{2(N-1)} \\ \alpha^0 & \alpha^3 & \alpha^6 & \ldots & \alpha^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha^0 & \alpha^{N-1} & \alpha^{2(N-1)} & \ldots & \alpha^{(N-1)^2} \end{bmatrix}$$

und für die Inverse Fourier-Transformation gilt:

$$\mathbf{F^{-1}} = \frac{1}{N}\cdot \begin{bmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \ldots & \alpha^0 \\ \alpha^0 & \alpha^{-1} & \alpha^{-2} & \ldots & \alpha^{-1(N-1)} \\ \alpha^0 & \alpha^{-2} & \alpha^{-4} & \ldots & \alpha^{-2(N-1)} \\ \alpha^0 & \alpha^{-3} & \alpha^{-6} & \ldots & \alpha^{-3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha^0 & \alpha^{-(N-1)} & \alpha^{-2(N-1)} & \ldots & \alpha^{-(N-1)^2} \end{bmatrix}$$