Unterstützende Materialien zur Vorlesung

Verfahren zur Kanalcodierung – Teil 3

Prof. Dr. Bernd Friedrichs
KIT CEL

Inhalt

- Schranken für die Minimaldistanz
- Asymptotische Schranken für die Minimaldistanz
- Wahrscheinlichkeit unerkannter Fehler bei Fehlererkennungscodes
- Fehlerwahrscheinlichkeit bei Hard-Decision
- Generatormatrix
- Prüfmatrix
- Hamming-Codes
- Einfache Modifikationen linearer Codes

Die Bedeutung von $d_{\min}$ und t sind nach dem vorangehenden Abschnitt klar. Wie hängen diese Größen aber mit den Codeparametern n, k, q zusammen?

Satz 3.7 (Singleton-Schranke). *Für einen linearen $(n, k, d_{\min})_q$-Code muß*

$$d_{\min} \leq n - k + 1 \tag{3.3.1}$$

gelten. Bei Gleichheit in der Singleton-Schranke liegt ein sogenannter MDS-Code *(Maximum Distance Separable) vor.*

Typ = Obere Schranke

Beweis: Alle Codewörter unterscheiden sich an mindestens $d_{\min}$ Stellen. Wenn bei allen Codewörtern die ersten $d_{\min} - 1$ Stellen gestrichen werden, so sind die gekürzten Codewörter der Länge $n - d_{\min} + 1$ immer noch alle verschieden. Es gibt also q^k verschiedene gekürzte Codewörter im Raum der $q^{n-d_{\min}+1}$ gekürzten Wörter. Dies ist aber nur möglich, wenn $k \leq n - d_{\min} + 1$ gilt. ■

Fazit:

Fehlererkennung: 1 Fehler erfordert 1 Prüfstelle: $t' = d_{min}-1 \leq n-k$

Fehlerkorrektur: 1 Fehler erfordert 2 Prüfstellen: $t = \lfloor (d_{min}-1)/2 \rfloor \leq (n-k)/2$

Satz 3.9 (Hamming-Schranke, sphere-packing bound). *Für einen linearen $(n, k, d_{\min})_q$-Code, der bis zu t Fehler korrigieren kann, muß*

Typ = Obere Schranke

$$q^{n-k} \geq \sum_{r=0}^{t} \binom{n}{r} (q-1)^r \quad \textit{bzw.} \quad n - k \geq \log_q \left[\sum_{r=0}^{t} \binom{n}{r} (q-1)^r \right] \tag{3.3.2}$$

gelten. Speziell für $q = 2$ bedeutet das:

$$2^{n-k} \geq \sum_{r=0}^{t} \binom{n}{r} = 1 + n + \binom{n}{2} + \cdots + \binom{n}{t}. \tag{3.3.3}$$

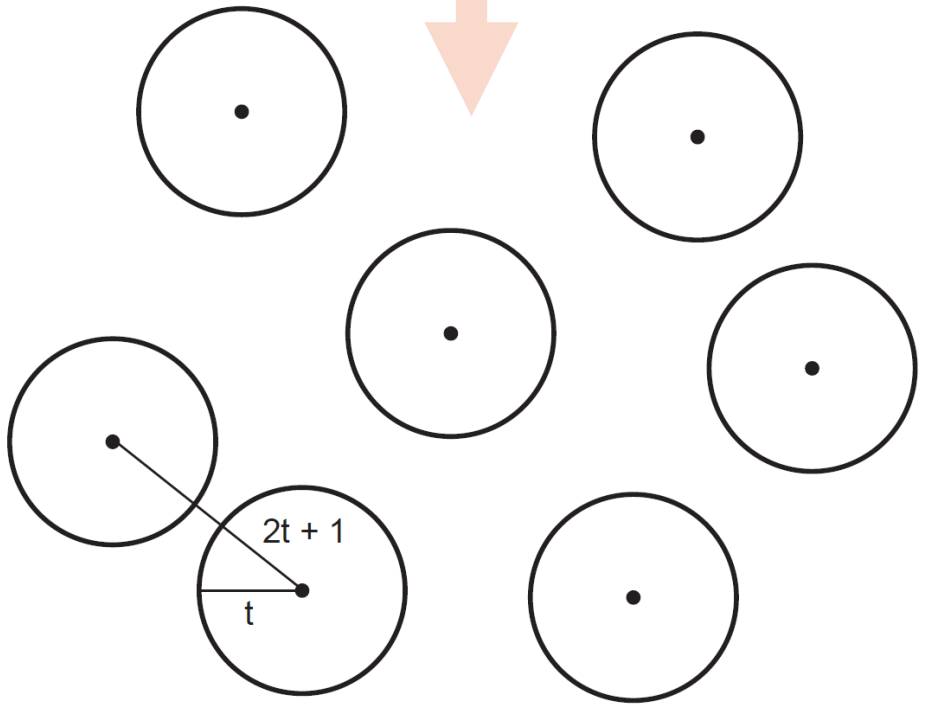
Ein sogenannter perfekter Code liegt vor, wenn eine Zahl t existiert, so daß Gleichheit in der Hamming-Schranke gilt. In diesem Fall sind die Decodierprinzipien MLD und BMD identisch.

Die Hamming-Schranke macht keine Aussagen über die Existenz von Codes! Wenn für eine Parameterkombination n, k, t, q die Hamming-Schranke erfüllt ist, dann wird damit noch längst nicht die Existenz eines entsprechenden Codes mit $d_{\min} \geq 2t + 1$ garantiert. Nur der umgekehrte Fall ist garantiert: Wenn die Parameterkombination der Hamming-Schranke nicht genügt, dann kann prinzipiell kein entsprechender Code existieren. Sinngemäß gilt dies auch für die Singleton-Schranke

Beweis: Die Decodierkugeln um die Codewörter sind disjunkt. Da es q^k Codewörter gibt, beträgt die Gesamtzahl aller Wörter in allen Decodierkugeln nach (3.2.3) genau

$$q^k \cdot \sum_{r=0}^{t} \binom{n}{r} (q-1)^r$$

und diese Zahl muß kleiner oder gleich sein als die Gesamtzahl q^n aller Wörter. Bei Gleichheit in der Hamming-Schranke sind die Decodierkugeln so dicht gepackt, daß sie den gesamten Raum $\mathbb{F}_q^n$ ausschöpfen. ■



$$\mathcal{C} = \{\ 0000000,\ 1000011,\ 0001111,\ 1001100,\ 0010110,\ 1010101,\ 0011001,\ 1011010,\ 0100101,\ 1100110,\ 0101010,\ 1101001,\ 0110011,\ 1110000,\ 0111100,\ 1111111\ \}.$$

Beispiel 3.6. **(1)** Der $(7,4,3)_2$-Hamming-Code aus Beispiel 1.2 mit $d_{\min} = 3$ und $t = 1$ ist perfekt, denn es gilt $2^{7-4} = 1+7$. Es gibt kein Wort, das von allen 16 Codewörtern einen Abstand ≥ 2 hat.

Nochmals: Jede Decodierkugel enthält 8 Wörter (Mittelpunkt und 7 Wörter mit Abstand 1),
Es gibt 16 Kugeln, die 8*16=128=2^7 Wörter füllen den gesamten Raum. Deshalb perfekt.

(2) Die Existenz des sogenannten $(23,12,7)_2$-Golay-Codes mit $t = 3$ wird hier nicht gezeigt. Einfach ist nur der Nachweis, daß ein solcher Code perfekt ist:

$$2^{23-12} = 2048 = 1 + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}.$$

(3) Betrachte einen $(127, 113, d)_2$-Code (der sich in Kapitel 7 als BCH-Code herausstellen wird). Gesucht ist d unter der Annahme, daß der Code bestmöglich konstruiert wurde. Aus der Hamming-Schranke folgt:

$$2^{127-113} = 16384 \geq \left\{ \begin{array}{lcll} 1 + 127 + \binom{127}{2} & = & 8129 & t = 2 \\ 1 + 127 + \binom{127}{2} + \binom{127}{3} & = & 341504 & t = 3 \end{array} \right\}.$$

Es folgt $t = 2$ und somit $d = 5$ oder $d = 6$. Bei 113 Infobits mit 14 Prüfbits können also 2 Fehler korrigiert werden oder es sind 4 (eventuell auch 5) Fehler erkennbar.

(4) Nach der Hamming-Schranke könnten ein $(20, 10)_2$-Code mit $t = 2$ und ein $(100, 50)_2$-Code mit $t = 11$ existieren, wobei die Coderate jeweils $1/2$ beträgt. Durch 5-fache Wiederholung kann aus dem $(20, 10)$-Code ebenfalls ein $(100, 50)$-Code konstruiert werden, indem die 10er-Abschnitte des 50er-Infowortes wie beim $(20, 10)$-Code encodiert werden. Der so entstehende Code kann aber weder 11 noch 10 Fehler korrigieren, sondern weiterhin nur 2 Fehler, denn 3 Fehler in einem 20er-Abschnitt sind nicht korrigierbar.

Gute Codes großer Blocklänge können also nicht mit dem Wiederholungsprinzip aus Codes kurzer Blocklänge erzeugt werden. ■

Der Begriff perfekter Code ist eigentlich übertrieben, da perfekte Codes von geringer praktischer Bedeutung sind. Perfekt sind nur die Hamming-Codes und Golay-Codes (aufwendiger Beweis) sowie:

Satz 3.10. *Der $(n,1)_2$-Wiederholungscode ist bei ungerader Blocklänge ein perfekter Code.*

Beweis: Sei $n = d_{\min} = 2t+1$ und sei $\boldsymbol{y} \in \mathbb{F}_2^n$ beliebig: Für $w_H(\boldsymbol{y}) \leq t$ gilt $d_H(\boldsymbol{y}, \boldsymbol{0}) = w_H(\boldsymbol{y}) \leq t$ und für $w_H(\boldsymbol{y}) \geq t+1$ gilt $d_H(\boldsymbol{y}, \boldsymbol{1}) = n - w_H(\boldsymbol{y}) \leq t$. Also folgt $K_t(\boldsymbol{0}) \cup K_t(\boldsymbol{1}) = \mathbb{F}_2^n$ und somit schöpfen die t-Kugeln um die Codewörter den Raum vollständig aus. ■

Die Singleton-, die Hamming- und die Plotkin-Schranke geben obere Grenzen für einen Code mit gewissen Eigenschaften an, wobei die Existenz eines solchen Codes nicht garantiert wird. Die folgende untere Schranke sichert dagegen die Existenz eines Codes, womit allerdings wie beim Kanalcodierungstheorem nicht die praktische Kenntnis des Codes verbunden ist:

Satz 3.12 (Gilbert-Varshamov-Schranke). *Es existiert immer ein linearer* $(n, k, d_{\min})_q$*-Code, sofern*

Typ = Untere Schranke

$$\sum_{r=0}^{d_{\min}-2} \binom{n-1}{r} (q-1)^r < q^{n-k} \qquad (3.3.5)$$

gilt. Für verschiedene weitere Formen dieser Schranke siehe z.B. [18, 115].

Obere vs. untere Schranke bezieht sich auf Existenz von Codes und hat nicht mit den Ungleichheitszeichen zu tun:

Obere Schranke: „wenn Code existiert dann gilt Schranke“ ⇔ „wenn Schranke verletzt ist dann existiert Code nicht“

d.h.: „wenn Schranke gilt dann folgt daraus nix“

Untere Schranke: „wenn Schranke gilt dann existiert Code“

Beispiel 3.7. Betrachte einen $(63, k, 5)_2$-Code, d.h. vorgegeben wird die Blocklänge 63 und es wird die Korrektur von 2 Fehlern verlangt und dabei sollen so wenig Prüfstellen wie möglich verwendet werden. Aus der Hamming-Schranke folgt:

$$\sum_{r=0}^{2} \binom{63}{r} = 1 + 63 + 1953 = 2017 \leq 2^{n-k}.$$

Also existiert eventuell ein Code mit nur 11 Prüfbits, d.h. $k \leq 52$. Aus der Gilbert-Varshamov-Schranke folgt:

$$\sum_{r=0}^{3} \binom{62}{r} = 1 + 62 + 1891 + 37820 = 39774 < 2^{n-k}.$$

Also ist die Existenz eines Codes gesichert mit nur 16 Prüfbits, d.h. $k \geq 47$. Tatsächlich gibt es einen $(63, 51, 5)_2$-BCH-Code (siehe Tabelle 7.1), d.h. dieser Code ist ziemlich gut und die Suche nach einem noch besseren Code lohnt hier kaum.

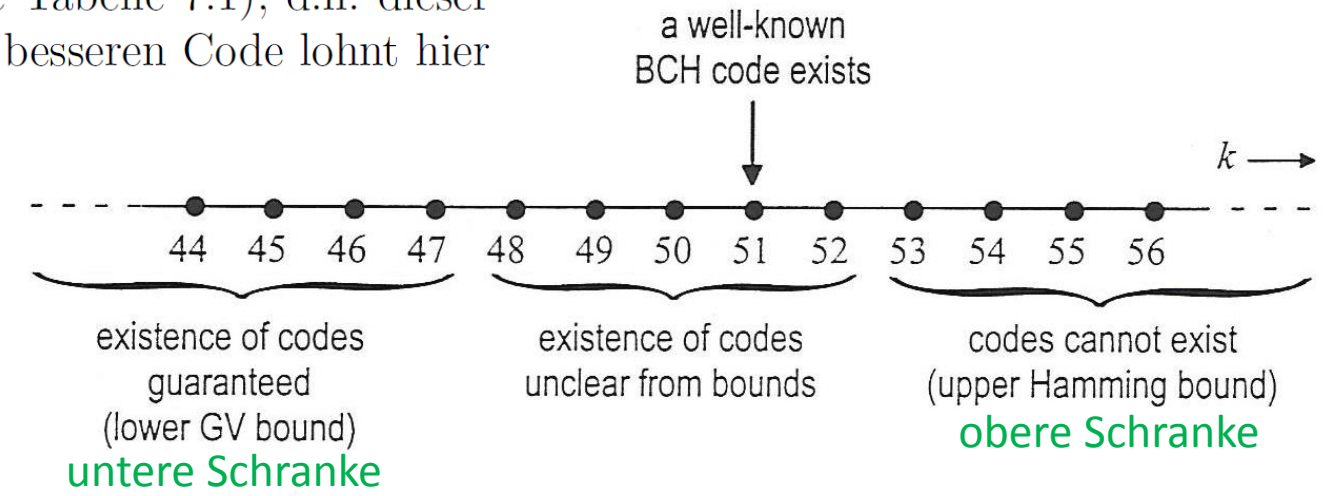


Figure 4.5. Gilbert-Varshamov and Hamming bounds for $(63, k, 5)_2$ codes

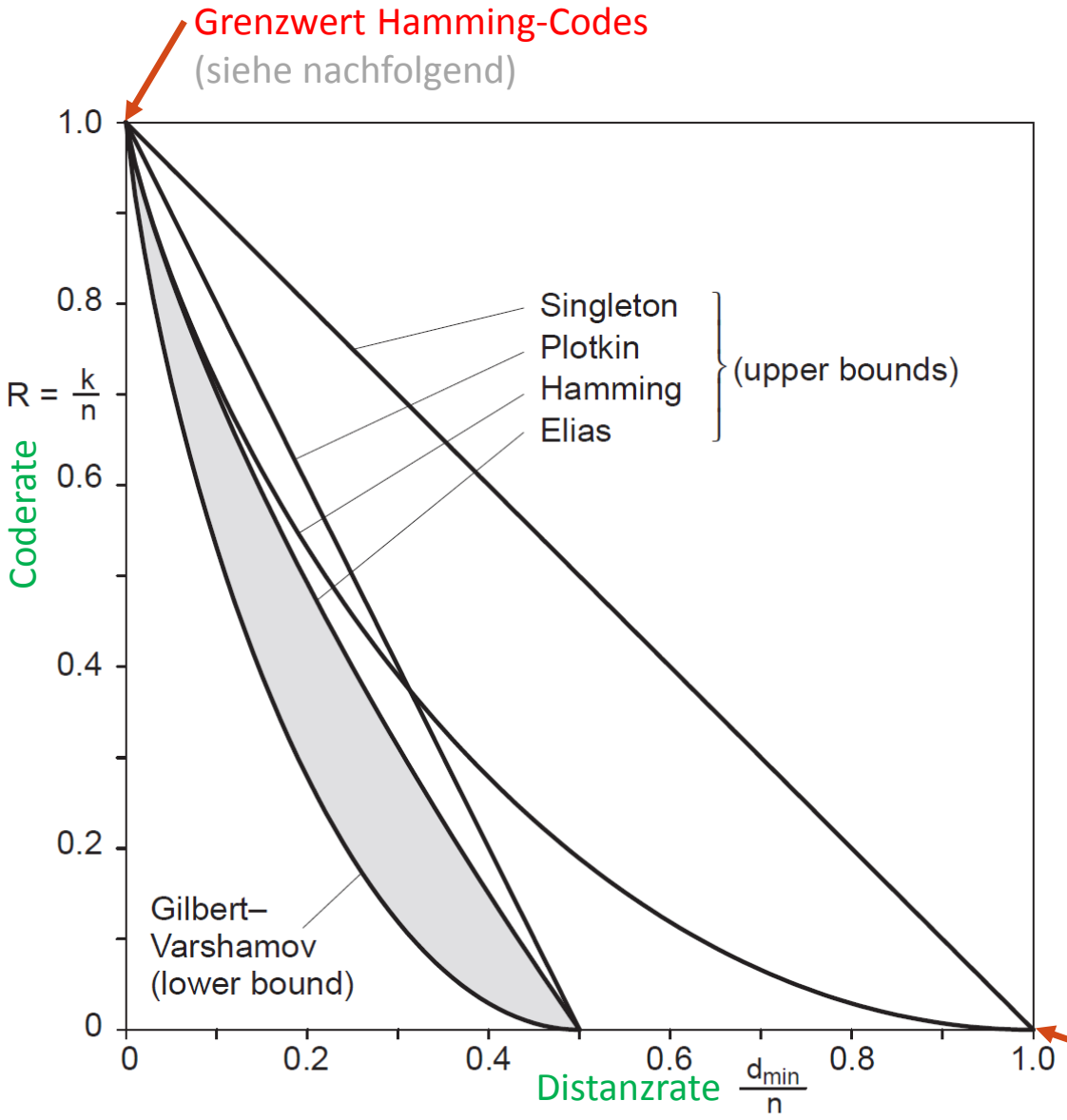


Bild 3.4. Asymptotische Schranken

Es wird jetzt der Fall $n \to \infty$ bei konstanter Coderate R betrachtet, so daß damit auch $k = Rn \to \infty$ impliziert wird. Für alle oberen bzw. unteren Schranken aus dem vorangehenden Abschnitt gibt es asymptotische Formen, bei denen die sogenannte *Distanzrate* $d_{\min}/n$ gegen einen Grenzwert bei $n \to \infty$ konvergiert. Bei den asymptotischen Schranken werden dann dieser Grenzwert der Distanzrate und die Coderate direkt miteinander verknüpft. Nachfolgend werden nur Binärcodes betrachtet.

Singleton-Schranke: Aus (3.3.1) folgt direkt $d_{\min}/n \le (n-k+1)/n \approx 1-R$ und somit:

$$R \le 1 - \frac{d_{\min}}{n}. \tag{3.4.1}$$

Hamming-Schranke: Aus (3.3.2) folgt direkt $1 - R \ge n^{-1} \log_2 \sum_{r=0}^{t} \binom{n}{r}$. Die rechte Seite dieser Ungleichung konvergiert nach Satz A.1 gegen die binäre Entropiefunktion, indem $\lambda = t/n \approx d_{\min}/(2n)$ gesetzt wird:

$$R \le 1 - H_2\left(\frac{d_{\min}}{2n}\right). \tag{3.4.2}$$

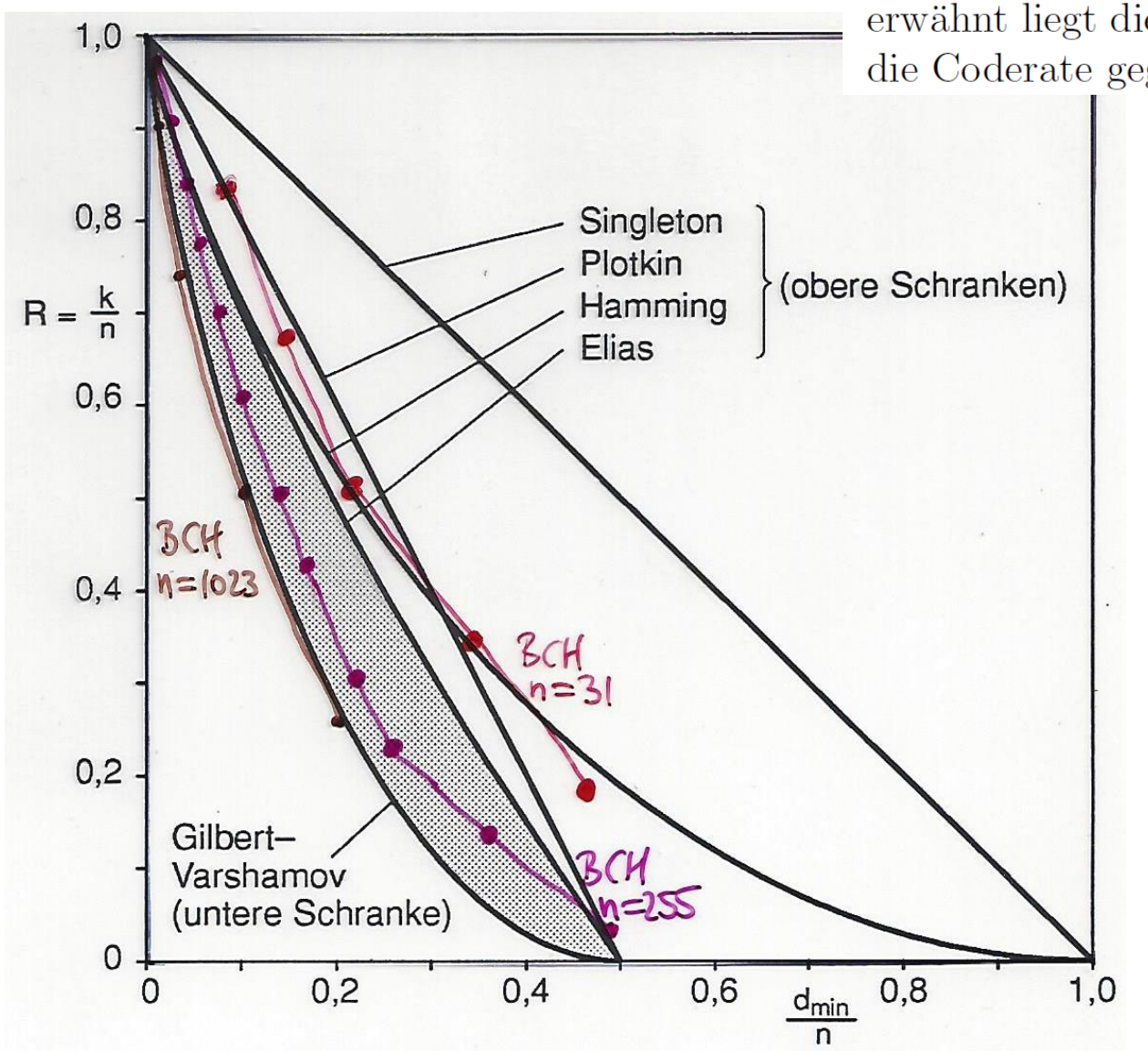
In Bild 3.4 erfolgt ein Vergleich der asymptotischen Schranken, indem die Coderate R als Funktion von $d_{\min}/n$ dargestellt wird. Alle Codes liegen unter den oberen Schranken, insbesondere also unter der Elias-Schranke, und es gibt einige gute Codes oberhalb der unteren Gilbert-Varshamov-Schranke, d.h. im schraffierten Bereich. Unterhalb des schraffierten Bereiches liegende Codes sind als schlecht anzusehen.

Nach der Gilbert-Varshamov-Schranke ist zumindest die Existenz von sogenannten *asymptotisch guten Codefamilien* (n_s, k_s, d_s) mit

$$\lim_{s\to\infty} \frac{k_s}{n_s} > 0 \quad \text{und} \quad \lim_{s\to\infty} \frac{d_s}{n_s} > 0 \tag{3.4.6}$$

garantiert. Dies erscheint auf den ersten Blick wenig bedeutsam und ganz selbstverständlich, aber alle bekannten Codefamilien (abgesehen von verketteten Systemen) erfüllen diese Eigenschaft nicht und sind damit *asymptotisch schlecht*,

Wie ist es eigentlich möglich, daß es Codes gibt, die die Singleton-Schranke (MDS-Codes) bzw. die Hamming-Schranke (perfekte Codes) annehmen, obwohl die kleinere Elias-Schranke das scheinbar ausschließt? Wie vorangehend bereits erwähnt liegt die Ursache darin, daß bei allen bekannten Codefamilien entweder die Coderate gegen Null oder die Distanzrate gegen Null konvergiert:



Eingezeichnet sind hier die Codes der BCH-Familie (Details später) zu den Blocklängen n=31, 255, 1023.

Je größer die Blocklänge, desto schlechter die Codes im Coderate-über-Distanzrate Diagramm. Also scheinen die BCH-Codes ungeeignet zu sein um das Shannon‘sche Kanalcodierungstheorem zu erfüllen.

Das ist aber nur ein Gesichtspunkt der BCH-Codes, der asymptotische Codierungsgewinn und der Verarbeitungsaufwand sind weitere Gesichtspunkte.

Grenzwerte in Diagramm der asymptotischen Schranken, Herleitung:

Details zu Hamming-Codes siehe später

Hamming-Codes sind nach Satz 4.10 von der Form

$$(n, k, d_{\min})_2 = (2^r - 1, 2^r - r - 1, 3)_2$$

und somit gilt $R \to 1$ und $d_{\min}/n \to 0$ für $r \to \infty$. Wiederholungscodes sind nach Satz 3.10 von der Form

$$(n, k, d_{\min})_2 = (2n + 1, 1, 2n + 1)_2$$

mit $R \to 0$ und $d_{\min}/n = 1$.

Zwar ist die Minimaldistanz der wichtigste Parameter eines Blockcodes, aber zur Berechnung der Fehlerwahrscheinlichkeit muß die gesamte sogenannte Gewichtsverteilung bekannt sein:

Definition 3.7. *Die Gewichtsverteilung (weight distribution) eines linearen $(n, k, d_{\min})_q$-Blockcodes ist ein Satz von Parametern $A_0, \ldots, A_n$, wobei A_r die Anzahl der Codewörter vom Hamminggewicht r bezeichnet. Der Gewichtsverteilung ist in umkehrbar eindeutiger Weise eine* Gewichtsfunktion *(weight enumerator) zugeordnet, wobei Z nur ein formaler Platzhalter ist:*

$$A(Z) = \sum_{r=0}^{n} A_r Z^r = \sum_{\boldsymbol{a} \in \mathcal{C}} Z^{w_H(\boldsymbol{a})}. \tag{3.5.1}$$

Es gelten folgende Eigenschaften:

$$A_0 = A(0) = 1 \quad , \quad A_n \leq (q-1)^n \tag{3.5.4}$$

$$A_r = 0 \text{ für } 0 < r < d_{\min} \tag{3.5.5}$$

$$\sum_{r=0}^{n} A_r = A(1) = q^k. \tag{3.5.6}$$

Beispiel 3.8. **(1)** Für den $(7, 4, 3)_2$-Hamming-Code aus Beispiel 1.2 ergibt sich $A_0 = A_7 = 1$ und $A_3 = A_4 = 7$ durch einfaches Abzählen. Die Gewichtsverteilung ist symmetrisch mit $A(Z) = 1 + Z^7 + 7(Z^3 + Z^4) = Z^7 \cdot A(Z^{-1})$.

(2) Für den $(n, 1, n)_2$-Wiederholungscode gilt offensichtlich $A(Z) = 1 + Z^n$.

Satz 3.14 (Fehlererkennung). *Vorausgesetzt wird ein linearer $(n, k, d_{\min})_q$-Code mit der Gewichtsverteilung $A_0, \ldots, A_n \leftrightarrow A(Z)$. Bei Übertragung über den q-nären symmetrischen Hard-Decision DMC mit der Symbol-Fehlerwahrscheinlichkeit p_e kann die Wahrscheinlichkeit P_{ue} (ue = undetected error) eines unentdeckbaren Fehlermusters exakt berechnet werden:*

$$P_{ue} = P(\boldsymbol{e} \in \mathcal{C} \backslash \{\boldsymbol{0}\}) = \sum_{r=d_{\min}}^{n} A_r \left(\frac{p_e}{q-1}\right)^r (1-p_e)^{n-r} \qquad (3.6.1)$$

Diese Formel erscheint simpel aber dennoch sind keine allgemein gültigen Aussagen ableitbar.

Im binären Fall wird P_{ue} normalerweise maximal bei $p_e = 1/2$ und ist dann durch die Anzahl der Prüfstellen exponentiell begrenzt:

$$P_{ue} \leq P_{ue}(p_e = 1/2) = \frac{2^k - 1}{2^n} \leq 2^{-(n-k)}. \qquad (3.6.4)$$

Das rechte ≤ gilt immer, das linke ≤ gilt bei improper Codes jedoch nicht.

Achtung: Es gibt aber auch sogenannte improper Codes mit $P_{ue} \gg 2^{-(n-k)}$ bei $p_e \ll 1/2$. Deshalb wird (3.6.4) auch als trügerische Schranke bezeichnet.

In technischen Anwendungen werden zur Fehlererkennung sogenannte CRC-Codes verwendet (siehe später), die noch weitere spezielle Eigenschaften aufweisen und garantiert nicht improper sind.

Beispiel 3.9. **(1)** Für den $(7,4,3)_2$-Hamming-Code aus Beispiel 3.8(1) gilt

$$P_{ue} = 7p_e^3(1-p_e)^4 + 7p_e^4(1-p_e)^3 + p_e^7 = 7p_e^3 - 21p_e^4 + 21p_e^5 - 7p_e^6 + p_e^7.$$

(2) Für den $(n,1,n)_2$-Wiederholungscode folgt aus der Definition von P_{ue} direkt $P_{ue} = P(\boldsymbol{e} = 11\ldots1) = p_e^n$. Das gleiche Ergebnis liefert $A(Z) = 1 + Z^n$:

$$P_{ue} = (1-p_e)^n \left(A\left(\frac{p_e}{1-p_e}\right) - 1 \right) = (1-p_e)^n \frac{p_e^n}{(1-p_e)^n} = p_e^n.$$

In den Bildern:
(1) Wiederholungscode
(2) $(7,3,4)_2$-Code
(3) $(7,4,3)_2$-Hamming-Code
(4) Parity-Check Code
(5) Improper Code, absichtlich ungünstig gewählt
(6) uncodiert

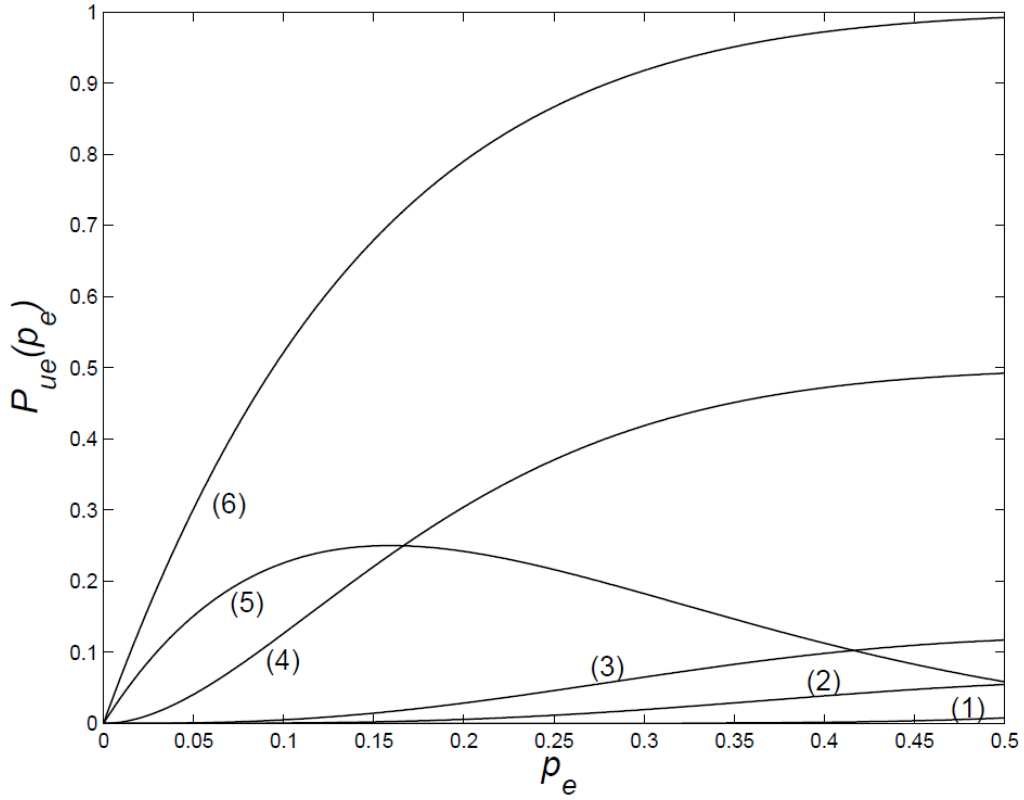


Figure 4.7a. Undetected error probability (linearly scaled axes)

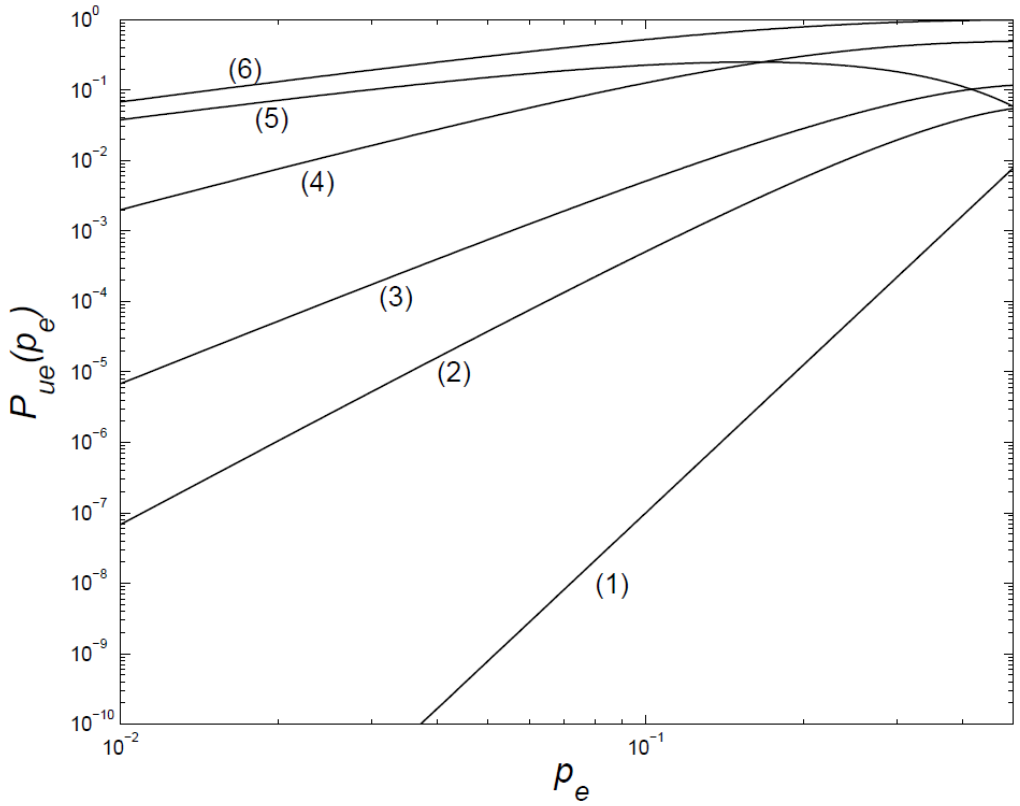


Figure 4.7b. Undetected error probability (both axes logarithmically scaled)

Beispiel: uncodierte Übertragung
wird beschrieben durch $(n, n, 1/2)$-Code

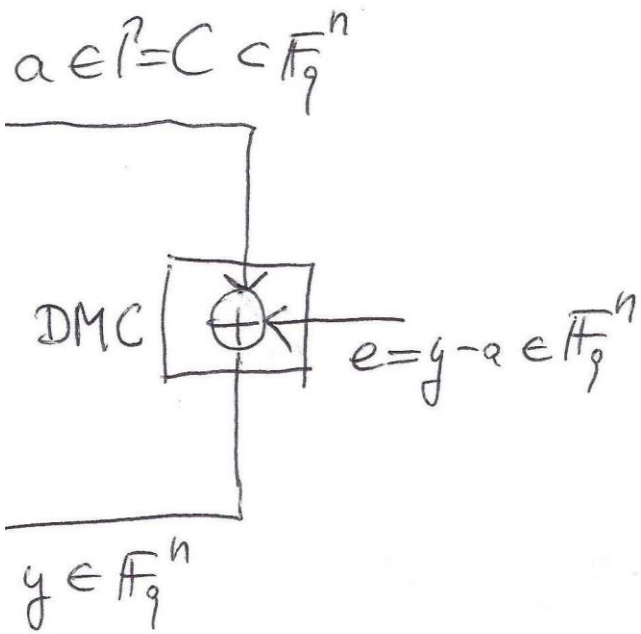
Gewichtsverteilung: $A_r = \binom{n}{r}$, check $\sum_{r=0}^{n} A_r = 2^n$

$$P_{ue}(p_e) = \sum_{r=1}^{n} \binom{n}{r} p_e^r (1-p_e)^{n-r}$$

$$= \underbrace{\sum_{r=0}^{n} \binom{n}{r} p_e^r (1-p_e)^{n-r}}_{=1} - \binom{n}{0} p_e^0 (1-p_e)^n$$

$$= 1 - (1-p_e)^n$$

$$= p_{ee} = P(e \neq 0) \qquad \text{siehe (1.3.6)}$$



Satz 3.15 (Fehlerkorrektur). *Vorausgesetzt wird ein linearer $(n, k, d_{\min})_q$-Code mit $t = \lfloor (d_{\min} - 1)/2 \rfloor$ und ML-Decodierung. Bei Übertragung über den q-nären symmetrischen Hard-Decision DMC mit der Symbol-Fehlerwahrscheinlichkeit p_e gilt für die Wort-Fehlerwahrscheinlichkeit P_w folgende Abschätzung:*

$$P_w \le 1 - \sum_{r=0}^{t} \binom{n}{r} p_e^r (1-p_e)^{n-r} = \sum_{r=t+1}^{n} \binom{n}{r} p_e^r (1-p_e)^{n-r}. \qquad (3.7.1)$$

Bei der Decodierung nach dem BMD-Prinzip oder bei perfekten Codes gilt hier sogar Gleichheit, d.h. P_w kann exakt berechnet werden. Für die Bit-Fehlerwahrscheinlichkeit P_b gilt allgemein die Abschätzung

$$P_b \le \sum_{r=t+1}^{n} \min\left\{1, \frac{r+t}{k}\right\} \binom{n}{r} p_e^r (1-p_e)^{n-r}. \qquad (3.7.2)$$

Für kleines p_e gelten näherungsweise die Abschätzungen:

$$P_w \lessapprox \binom{n}{t+1} p_e^{t+1}, \quad P_b \lessapprox \min\left\{1, \frac{d_{\min}}{k}\right\} \cdot P_w. \qquad (3.7.3)$$

Beweis: Da das MLD-Prinzip besser als das BMD-Prinzip ist, braucht nur die Gleichheit für BMD bewiesen zu werden. Eine korrekte BM-Decodierung erfolgt genau dann, wenn maximal t Fehler auftreten:

$$\begin{aligned} P_w &= 1 - P(\text{richtige Decodierung}) \\ &= 1 - P(w_H(\boldsymbol{e}) \le t) \;=\; P(w_H(\boldsymbol{e}) \ge t+1) \\ &= 1 - \sum_{r=0}^{t} P(w_H(\boldsymbol{e}) = r) \;=\; \sum_{r=t+1}^{n} P(w_H(\boldsymbol{e}) = r). \end{aligned}$$

Die Anzahl der Fehler in einem Wort der Länge n ist nach (1.3.9) binomialverteilt:

$$P(w_H(\boldsymbol{e}) = r) \;=\; \binom{n}{r} p_e^r (1-p_e)^{n-r}.$$

Damit ist (3.7.1) bewiesen, wobei die Gleichheit in (3.7.1) auch direkt aus der binomischen Formel (A.2.2) folgt.

Für kleines p_e dominiert der erste Summand im rechten Term von (3.7.1). Für P_b gilt dabei $(r+t)/k = (2t+1)/k = d_{\min}/k$. Für größeres p_e werden die Fehlerwahrscheinlichkeiten mit (3.7.3) aber eventuell unterschätzt, so daß sich die obere Schranke in eine untere Schranke verwandeln kann. ■

Achtung: Schon bei einigermaßen kleinen Werten von p_e kann die Differenz im linken Term von (3.7.1) numerisch kaum ausgewertet werden, so daß in diesem Fall immer der rechte Term benutzt werden sollte.

Mit den Resultaten aus Satz 3.15 wurden die Bit- und Wort-Fehlerwahrscheinlichkeiten in den Bildern 1.10, 1.11, 3.5 sowie die BCH-Kurven in Abschnitt 7.3 berechnet.

Beispiel 3.10. (1) Der $(7,4,3)_2$-Hamming-Code mit $t = 1$ ist nach Beispiel 3.6(1) perfekt und somit gilt nach (3.7.1):

$$\begin{aligned} P_w &= 1 - \binom{7}{0} p_e^0 (1-p_e)^7 - \binom{7}{1} p_e^1 (1-p_e)^6 \;=\; 1 - (1-p_e)^7 - 7p_e(1-p_e)^6 \\ &= 1 - (1 - 7p_e + 21p_e^2 - p_e^3 \ldots) - 7p_e(1 - 6p_e + p_e^2 \ldots) \\ &\approx 21p_e^2 \;=\; \binom{7}{2} p_e^2. \end{aligned}$$

Damit wurde die Näherung (3.7.3) bestätigt. Der asymptotische Codierungsgewinn beträgt $G_{\text{a,hard}} = 10 \cdot \log_{10}(4/7 \cdot 2) = 0{,}6$ dB.

Satz 3.17 (AWGN). *Vorausgesetzt wird ein linearer $(n, k, d_{\min})_2$-Code mit der Gewichtsverteilung $A_0, \ldots, A_n$ und ML-Decodierung. Bei der Übertragung über den AWGN mit $q = 2$ und idealer Soft-Decision gilt für die Wort-Fehlerwahrscheinlichkeit P_w folgende Abschätzung:*

$$P_w \le \sum_{r=d_{\min}}^{n} A_r Q\left(\sqrt{2r\frac{E_c}{N_0}}\right) = \sum_{r=d_{\min}}^{n} A_r Q\left(\sqrt{2Rr\frac{E_b}{N_0}}\right). \tag{3.8.5}$$

Für einen guten Kanal mit großem E_b/N_0 gilt näherungsweise:

$$P_w \le A_{d_{\min}} Q\left(\sqrt{2Rd_{\min}\frac{E_b}{N_0}}\right). \tag{3.8.6}$$

Für $E_b/N_0 \to \infty$ gilt (3.8.6) sogar asymptotisch exakt.

Nach (3.8.6) und (A.3.18) gilt für den AWGN mit Soft-Decision näherungsweise $P_b \approx \text{const} \cdot P_w \approx \text{const} \cdot e^{-Rd_{\min} \cdot E_b/N_0}$ und das wurde in (1.7.10) benutzt zur Herleitung des asymptotischen Codierungsgewinns mit dem Ergebnis $G_{\text{a,soft}} = 10 \cdot \log_{10}(Rd_{\min})$ dB.

Vorausgesetzt wird ein linearer $(n,k)_q$-Code $\mathcal{C}$. Nach Definition 3.3 ist die Codemenge $\mathcal{C}$ ein Vektorraum mit q^k Wörtern bzw. Vektoren. $\mathcal{C}$ kann auch als Untervektorraum des Vektorraums $\mathbb{F}_q^n$ aller q^n möglichen Wörter aufgefaßt werden. Für die Grundbegriffe zu Vektorräumen wird auf Abschnitt A.5 verwiesen. Insbesondere ist jede *Linearkombination* von Codewörtern wieder ein Codewort, d.h.

$$\boldsymbol{a}_1, \ldots, \boldsymbol{a}_l \in \mathcal{C}, \quad \alpha_1, \ldots, \alpha_l \in \mathbb{F}_q \quad \Longrightarrow \quad \sum_{i=1}^{l} \alpha_i \boldsymbol{a}_i \in \mathcal{C}.$$

Die maximale Anzahl der linear unabhängigen Wörter entspricht der *Dimension* des Vektorraums bzw. des Codes und wird als $\mathrm{Dim}(\mathcal{C})$ geschrieben. Klar ist $\mathrm{Dim}(\mathcal{C}) \leq n$ und weiter gilt sogar $\mathrm{Dim}(\mathcal{C}) = k$, da ein Vektorraum der Dimension k über $\mathbb{F}_q$ genau q^k Wörter enthält. Jede Auswahl von $\mathrm{Dim}(\mathcal{C})$ linear unabhängigen Wörtern bildet eine *Basis* für den Code.

Mit $\mathbb{F}_q^n$ werden die Wörter bzw. Vektoren der Länge n mit Elementen aus $\mathbb{F}_q$ bezeichnet. Entsprechend steht $\mathbb{F}_q^{k,n}$ für die Menge der (k,n)-dimensionalen Matrizen mit Elementen aus $\mathbb{F}_q$.

Definition 4.1. *Eine Matrix* $\boldsymbol{G} \in \mathbb{F}_q^{k,n}$ *heißt* Generatormatrix *für den linearen* $(n,k)_q$*-Code* $\mathcal{C}$*, wenn gilt:*

$$\mathcal{C} = \left\{ \boldsymbol{uG} \;\middle|\; \boldsymbol{u} \in \mathbb{F}_q^k \right\}. \tag{4.1.1}$$

Die Generatormatrix erzeugt den Code und liefert gleichzeitig eine Encodiervorschrift, indem die Codewörter in folgender Form erzeugt werden:

$$\begin{aligned}(a_0, \ldots, a_{n-1}) &= (u_0, \ldots, u_{k-1}) \cdot \begin{pmatrix} g_{0,0} & \cdots & g_{0,n-1} \\ \vdots & & \vdots \\ g_{k-1,0} & \cdots & g_{k-1,n-1} \end{pmatrix} \\ &= (u_0 g_{0,0} + \cdots + u_{k-1} g_{k-1,0}, \ldots, u_0 g_{0,n-1} + \cdots + u_{k-1} g_{k-1,n-1}) \\ &= u_0 (g_{0,0}, \ldots, g_{0,n-1}) + \cdots + u_{k-1} (g_{k-1,0}, \ldots, g_{k-1,n-1}).\end{aligned}$$

Die Zeilen der Generatormatrix sollen linear unabhängig sein und bilden deshalb eine Basis für $\mathcal{C}$ *mit* $\mathrm{Dim}(\mathcal{C}) = k$.

Die Zeilen der Generatormatrix sind offensichtlich Codewörter zu den Einheitsvektoren als Infowörter. Jeder von einer Generatormatrix erzeugte Code ist linear, denn aus $\boldsymbol{a}_i = \boldsymbol{u}_i \boldsymbol{G}$ folgt:

$$\sum_{i=1}^{l} \alpha_i \boldsymbol{a}_i \;=\; \sum_{i=1}^{l} \alpha_i \Big(\boldsymbol{u}_i \boldsymbol{G} \Big) \;=\; \underbrace{\Big(\sum_{i=1}^{l} \alpha_i \boldsymbol{u}_i \Big)}_{\boldsymbol{u}} \cdot \boldsymbol{G} \;=\; \boldsymbol{u} \cdot \boldsymbol{G}.$$

Der Zeilenrang (Spaltenrang) einer Matrix ist die Anzahl der linear unabhängigen Zeilen (Spalten) bzw. die Dimension des davon erzeugten Vektorraums. Zeilenrang und Spaltenrang stimmen überein und werden deshalb kurz *Rang* genannt. Wegen $n > k$ sind die Spalten der Generatormatrix natürlich linear abhängig.

Beispiel 4.1. Betrachte den $(7,4)_2$-Hamming-Code mit der aufzählenden Beschreibung von $\mathcal{C}$ gemäß Beispiel 1.2. Eine passende Generatormatrix ist

$$\boldsymbol{G} = \left(\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right)$$

mit dem systematischen Encoder

$$(u_0, u_1, u_2, u_3) \mapsto (u_0, u_1, u_2, u_3, u_1 + u_2 + u_3, u_0 + u_2 + u_3, u_0 + u_1 + u_3).$$

Zahlenbeispiel:

$$(1,1,0,1) \mapsto (1,1,0,1,0+1+1,1+0+1,1+1+1) = (1,1,0,1,0,0,1).$$

Satz 4.1 (Elementare Zeilenoperationen). *Es sei $\boldsymbol{G}$ eine Generatormatrix für den $(n,k)_q$-Code $\mathcal{C}$. Dann sind in $\boldsymbol{G}$ die folgenden sogenannten* elementaren Zeilenoperationen *erlaubt, ohne daß sich der Code dadurch ändert:*

(1) Vertauschung zweier Zeilen.
(2) Multiplikation einer Zeile mit einem Skalar ungleich Null.
(3) Addition einer mit einem Skalar multiplizierten Zeile zu einer anderen Zeile.

Also erzeugen die vier folgenden Generatormatrizen jeweils den gleichen Code:

$$\begin{pmatrix} \vdots \\ \boldsymbol{g}_i \\ \vdots \\ \boldsymbol{g}_j \\ \vdots \end{pmatrix} \quad \begin{pmatrix} \vdots \\ \boldsymbol{g}_j \\ \vdots \\ \boldsymbol{g}_i \\ \vdots \end{pmatrix} \quad \begin{pmatrix} \vdots \\ \alpha\boldsymbol{g}_i \\ \vdots \\ \boldsymbol{g}_j \\ \vdots \end{pmatrix} \quad \begin{pmatrix} \vdots \\ \boldsymbol{g}_i \\ \vdots \\ \boldsymbol{g}_j + \alpha\boldsymbol{g}_i \\ \vdots \end{pmatrix}. \tag{4.1.2}$$

Mit diesen elementaren Zeilenoperationen (sowie eventuell zusätzlicher Spaltenvertauschungen) kann $\boldsymbol{G}$ überführt werden in die sogenannte Zeilennormalform *(Gaußsche Normalform, kanonische Staffelform, row echelon form):*

$$\boldsymbol{G} = \left(\boldsymbol{E}_k \,\middle|\, \boldsymbol{P}\right). \tag{4.1.3}$$

Dabei ist $\boldsymbol{E}_k \in \mathbb{F}_q^{k,k}$ eine Einheitsmatrix und $\boldsymbol{P} \in \mathbb{F}_q^{k,n-k}$.

Die Zeilennormalform entspricht einem systematischen Encoder, der also immer bei Blockcodes erreicht werden kann.

Beispiel 4.2. Betrachte wieder den $(7,4)_2$-Hamming-Code. Nach den Beispielen 4.1 oder 1.2 sind 1111111, 1011010, 0110011, 1110000 jeweils Codewörter. Wenn diese Codewörter linear unabhängig sind (was per Augenschein nicht sofort entscheidbar ist), dann bilden diese Wörter ebenfalls eine Basis und die damit gebildete Generatormatrix $\boldsymbol{G}_1$ muß einen identischen Code erzeugen:

$$\boldsymbol{G}_1 = \begin{pmatrix} 1&1&1&1&1&1&1 \\ 1&0&1&1&0&1&0 \\ 0&1&1&0&0&1&1 \\ 1&1&1&0&0&0&0 \end{pmatrix}.$$

Durch die elementaren Zeilenoperationen wird nun versucht, $\boldsymbol{G}_1$ in $\boldsymbol{G}$ aus Beispiel 4.1 zu überführen: Addiere Zeile 1 zu Zeile 2 und zu Zeile 4:

$$\boldsymbol{G}_2 = \begin{pmatrix} 1&1&1&1&1&1&1 \\ 0&1&0&0&1&0&1 \\ 0&1&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{pmatrix}.$$

Addiere Zeile 2 zu Zeile 1 und zu Zeile 3:

$$\boldsymbol{G}_3 = \begin{pmatrix} 1&0&1&1&0&1&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&1&1&0 \\ 0&0&0&1&1&1&1 \end{pmatrix}.$$

Addiere Zeile 3 zu Zeile 1:

$$\boldsymbol{G}_4 = \begin{pmatrix} 1&0&0&1&1&0&0 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&1&1&0 \\ 0&0&0&1&1&1&1 \end{pmatrix}.$$

Addiere Zeile 4 zu Zeile 1:

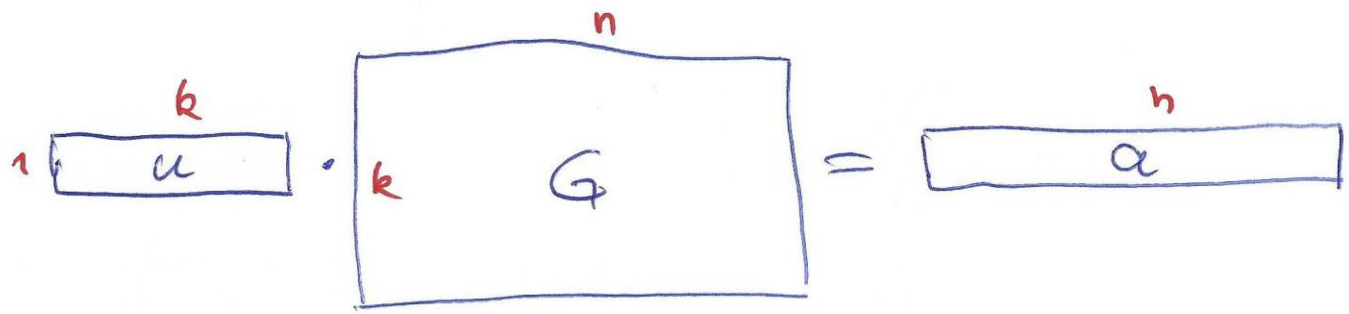
$$\boldsymbol{G}_5 = \left(\begin{array}{cccc|ccc} 1&0&0&0&0&1&1 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&1&1&0 \\ 0&0&0&1&1&1&1 \end{array}\right).$$

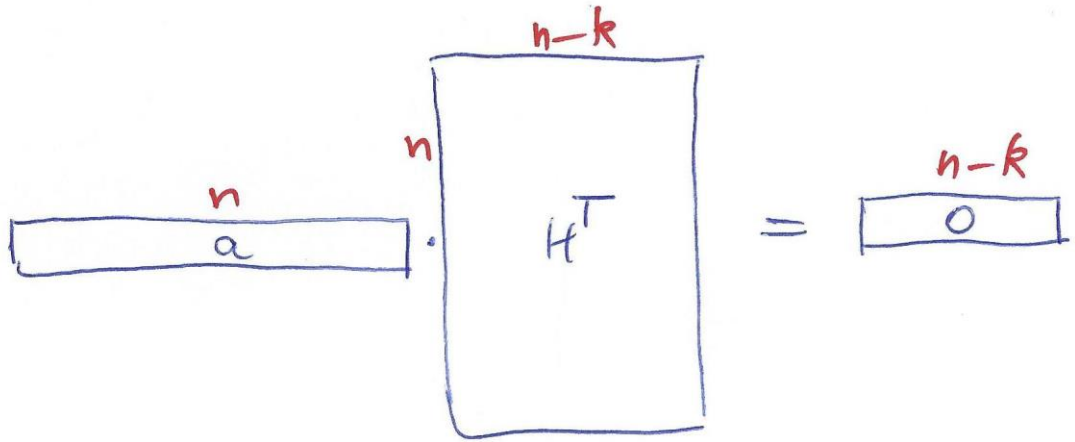
Dies ist wieder die originale Generatormatrix aus Beispiel 4.1. ■

Offensichtlich ist die Minimaldistanz kleiner oder gleich dem minimalen Hamminggewicht aller Zeilen der Generatormatrix, da die Zeilen Codewörter sind. Das folgende Beispiel zeigt aber, daß $d_{\min}$ auch echt kleiner als das minimale Zeilengewicht sein kann:

$$\boldsymbol{G} = \begin{pmatrix} 1&1&1&0 \\ 0&1&1&1 \end{pmatrix} \quad \text{erzeugt} \quad \mathcal{C} = \{0000, 1110, 0111, 1001\}.$$

Ein Code kann nicht nur über die Generatormatrix $\boldsymbol{G}$ definiert werden, sondern auch über die nachfolgend erklärte Prüfmatrix $\boldsymbol{H}$. Eine mögliche Prüfmatrix und eine mögliche Generatormatrix können jeweils auseinander hergeleitet werden. Ferner wird über die Prüfmatrix in Abschnitt 4.6 das zentrale Konzept des Syndroms eingeführt.





Definition 4.2. *Eine Matrix* $\boldsymbol{H} \in \mathbb{F}_q^{n-k,n}$ *heißt* Prüfmatrix *(Parity Check Matrix) für den linearen* $(n,k)_q$*-Code* $\mathcal{C}$*, wenn gilt:*

$$\mathcal{C} = \left\{ \boldsymbol{a} \in \mathbb{F}_q^n \;\middle|\; \boldsymbol{a}\boldsymbol{H}^T = \boldsymbol{0} \right\}. \tag{4.2.1}$$

Dabei hat das Nullwort die Länge $n-k$*. Für die Codewörter* $\boldsymbol{a} \in \mathcal{C}$ *gilt also* $\boldsymbol{a}\boldsymbol{H}^T = \boldsymbol{0}$ *und für alle anderen Wörter* $\boldsymbol{a} \notin \mathcal{C}$ *gilt* $\boldsymbol{a}\boldsymbol{H}^T \neq \boldsymbol{0}$*.* $\mathcal{C}$ *heißt auch* Nullraum *von* $\boldsymbol{H}$ *bzw.* Zeilenraum *von* $\boldsymbol{G}$*.*

Wie die Generatormatrix wird auch die Prüfmatrix durch den Code keinesfalls eindeutig bestimmt:

Satz 4.2. *Die Prüfmatrix* $\boldsymbol{H}$ *hat den maximal möglichen Rang* $n-k$ *und es sind die elementaren Zeilenoperationen für* $\boldsymbol{H}$ *erlaubt – aber nicht für* $\boldsymbol{H}^T$*!*

Satz 4.3. *Der lineare $(n,k)_q$-Code $\mathcal{C}$ werde erzeugt durch die Generatormatrix $\boldsymbol{G} \in \mathbb{F}_q^{k,n}$. Dann gilt:*

(1) Die Matrix $\boldsymbol{H} \in \mathbb{F}_q^{n-k,n}$ ist eine Prüfmatrix für $\mathcal{C}$ genau dann, wenn gilt:

$$\boldsymbol{H} \neq \boldsymbol{0} \quad \text{und} \quad \boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}. \tag{4.2.2}$$

(2) Wenn $\boldsymbol{G} = \left(\boldsymbol{E}_k \,\middle|\, \boldsymbol{P}\right)$ eine systematische Generatormatrix mit der Einheitsmatrix $\boldsymbol{E}_k \in \mathbb{F}_q^{k,k}$ und der Matrix $\boldsymbol{P} \in \mathbb{F}_q^{k,n-k}$ ist, dann wird eine Prüfmatrix gegeben durch

$$\boldsymbol{H} = \left(-\boldsymbol{P}^T \,\middle|\, \boldsymbol{E}_{n-k}\right). \tag{4.2.3}$$

Beispiel 4.3. **(1)** Für den $(7,4)_2$-Hamming-Code ist $\boldsymbol{G}$ aus Beispiel 4.1 bekannt und $\boldsymbol{H}$ wird gemäß (4.2.3) gebildet:

$$\boldsymbol{G} = \left(\begin{array}{cccc|ccc} 1&0&0&0&0&1&1\\ 0&1&0&0&1&0&1\\ 0&0&1&0&1&1&0\\ 0&0&0&1&1&1&1 \end{array}\right), \quad \boldsymbol{H} = \left(\begin{array}{cccc|ccc} 0&1&1&1&1&0&0\\ 1&0&1&1&0&1&0\\ 1&1&0&1&0&0&1 \end{array}\right).$$

(2) Für den $(n,1)_2$-Wiederholungscode ergibt sich $\boldsymbol{G}$ direkt und daraus folgt $\boldsymbol{H}$:

$$\boldsymbol{G} = \left(\begin{array}{c|ccccc} 1 & 1 & 1 & \dots & 1 & 1 \end{array}\right) \quad , \quad \boldsymbol{H} = \left(\begin{array}{c|ccccc} 1 & 1 & & & & \\ 1 & & 1 & & & \\ \vdots & & & \ddots & & \\ 1 & & & & 1 & \\ 1 & & & & & 1 \end{array}\right).$$

(3) Für den $(n,n-1)_2$-Parity Check Code ergibt sich $\boldsymbol{H}$ direkt und daraus folgt $\boldsymbol{G}$:

$$\boldsymbol{G} = \left(\begin{array}{c|ccccc} 1 & 1 & & & & \\ 1 & & 1 & & & \\ \vdots & & & \ddots & & \\ 1 & & & & 1 & \\ 1 & & & & & 1 \end{array}\right) \quad , \quad \boldsymbol{H} = \left(\begin{array}{c|ccccc} 1 & 1 & 1 & \dots & 1 & 1 \end{array}\right).$$

Mit $\boldsymbol{G}$ wird eine Encodierung bewirkt, bei der das Prüfbit vorangestellt ist. Aus dem Code selbst ist die Encodierung nicht ablesbar (siehe dazu auch Beispiel 1.1). ■

Eine einfache Berechnung der Minimaldistanz aus der Generatormatrix ist wie erwähnt nicht möglich. Jedoch gilt folgender Zusammenhang, der bereits beim Beweis der Gilbert-Varshamov-Schranke aus Satz 3.12 angewendet wurde:

Satz 4.4. *Es sei* $\boldsymbol{H} \in \mathbb{F}_q^{n-k,n}$ *eine Prüfmatrix für den* $(n, k, d_{\min})_q$*-Code* $\mathcal{C}$*. Dann ist die Minimaldistanz* $d_{\min}$ *die minimale Anzahl der linear abhängigen Spalten in* $\boldsymbol{H}$*, d.h.:*

Jede Auswahl von $d_{\min} - 1$ *Spalten ist linear unabhängig und es gibt mindestens eine Auswahl von* $d_{\min}$ *linear abhängigen Spalten.*

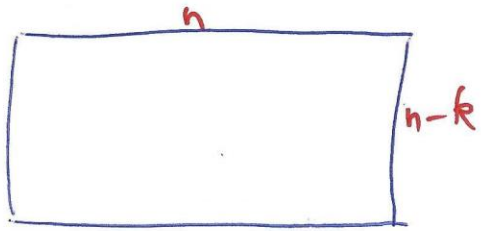
Dieser Satz bildet die Grundlage für die Herleitung der Hamming-Codes

Aus diesem Satz ergibt sich übrigens direkt die Singleton-Schranke aus Satz 3.7: Da die Spalten der Prüfmatrix die Länge $n - k$ haben, kann es maximal $n - k$ linear unabhängige Spalten geben. Also folgt $d_{\min} - 1 \leq n - k$.

Satz 4.4 lautet für $d_{min} = 3$

Jede Auswahl von $d_{min} - 1 = 2$ Spalten ist lin. unabh. und es gibt eine Auswahl von 3 lin. abh. Spalten

$\Longleftrightarrow$ Alle Spalten in H sind verschieden, keine Spalte ist Null

$H =$ 

Also:

$n \leq 2^{n-k} - 1 =$ Anzahl der versch. Spalten $\neq 0$ der Länge $n-k$

Speziell $n = 2^r - 1 \leq 2^{n-k} - 1$

$\Longrightarrow$ $n - k = r$ ist möglich, also $k = n - r$

Tatsächlich existiert die Familie der Hamming-Codes mit

$(2^r - 1,\ 2^r - r - 1,\ 3)_2$

Siehe §4.4

Die Codes sind perfekt:

$$2^{n-k} \overset{!}{=} \sum_{i=0}^{t} \binom{n}{i}, \quad t = 1$$

$2^{n-k} = 2^r$

$\sum_{i=0}^{t} \binom{n}{i} = 1 + n$

Definition 4.3. *Zum $(n,k)_q$-Code $\mathcal{C}$ gehöre die Generatormatrix $\boldsymbol{G} \in \mathbb{F}_q^{k,n}$ und die Prüfmatrix $\boldsymbol{H} \in \mathbb{F}_q^{n-k,n}$:*

Durch $\boldsymbol{H}$ als Generatormatrix bzw. $\boldsymbol{G}$ als Prüfmatrix wird ein $(n, n-k)_q$-Code erzeugt, der als dualer Code $\mathcal{C}^\perp$ bezeichnet wird.

Für beliebige $\boldsymbol{a} = \boldsymbol{uG} \in \mathcal{C}$ und $\boldsymbol{b} = \boldsymbol{vH} \in \mathcal{C}^\perp$ ist das Skalarprodukt Null, denn es gilt $\boldsymbol{ab}^T = \boldsymbol{uGH}^T\boldsymbol{v}^T = \boldsymbol{u0v}^T = 0$.

Beispiel 4.5. **(1)** Aufgrund von Beispiel 4.3(2,3) sind der $(n,1)_2$-Wiederholungscode und der $(n, n-1)_2$-Parity Check Code dual zueinander.

(2) Dual zum $(7,4)_2$-Hamming-Code $\mathcal{C}$ ist der durch $\boldsymbol{H}$ aus Beispiel 4.3(1) generierte $(7,3)_2$-Code

$$\begin{aligned}\mathcal{C}^\perp = \{\ & 0000\ 000, \quad 0111\ 100,\\ & 1101\ 001, \quad 1010\ 101,\\ & 1011\ 010, \quad 1100\ 110,\\ & 0110\ 011, \quad 0001\ 111\ \}.\end{aligned}$$

Offensichtlich sind die 16 Wörter aus $\mathcal{C}$ und die 8 Wörter aus $\mathcal{C}^\perp$ jeweils orthogonal zueinander. ■

Bisher wurde nur der $(7,4)_2$-Hamming-Code behandelt. Die Hamming-Codes bilden jedoch eine ganze Klasse von 1-Fehler-korrigierenden bzw. 2-Fehler-erkennenden Codes mit einer gegen 1 konvergierenden Coderate:

Satz 4.10. *Ein* $(n, k, d_{\min})_q = (n, n-r, 3)_q$*-Hamming-Code der Ordnung* r *ist durch*

$$n = \frac{q^r - 1}{q - 1} = 1 + q + q^2 + \cdots + q^{r-1} \tag{4.4.1}$$

definiert. Hamming-Codes existieren für alle Ordnungen und sind perfekt. Bis auf Permutationen (d.h. Spaltenvertauschungen bzw. Äquivalenzen) ist der Code eindeutig bestimmt. Nur für $r = 2$ *liegt ein MDS-Code vor.*

Speziell für $q = 2$ *liegt ein* $(2^r - 1, 2^r - r - 1, 3)_2$*-Code vor, Beispiele sind also*

$$(3,1),\ (7,4),\ (15,11),\ (31,26),\ (63,57),\ \ldots$$

In diesem Fall enthält die Prüfmatrix als Spalten die $2^r - 1$ *verschiedenen Binärwörter der Länge* r *(abgesehen vom Nullwort).*

Beispiel 4.8. **(1)** Die Prüfmatrix des $(7, 4, 3)_2$-Hamming-Codes der Ordnung $r = 3$ aus Beispiel 4.3(1) ist entsprechend Satz 4.10 konstruiert.

(2) Für den $(15, 11, 3)_2$-Hamming-Code der Ordnung $r = 4$ kann $\boldsymbol{H}$ wie folgt gewählt werden:

$$\boldsymbol{H} = \left(\begin{array}{cccccccccccc|cccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

Es gibt genau 15 Binärspalten der Länge 4, wenn von der Nullspalte abgesehen wird. Die vier Einheitsvektoren sind rechts zur Einheitsmatrix zusammengefaßt und die restlichen 11 Spalten sind als Dualzahlen geordnet. ■

The colored hats puzzle is a nice example for the successful application of Hamming codes to problems which do not seem to be connected to communications or coding. This puzzle was first published as an article which appeared in the Science Times section of the New York Times of April 10th, 2001.

The puzzle is stated as follows: Each player of a team is randomly and independently assigned to wear a colored hat (either red=0 or blue=1). Each player views the colors of his other teammates (but can not see his own color), and then tries to guess the color of his own hat. No communication is allowed between the players except for a strategy session before the game begins. It is allowed that some players do not guess and remain neutral. The team wins a prize if at least one player guesses his own color correctly and no player guesses incorrectly. Vice versa, the team loses if there are no guesses or at least one player guesses incorrectly.

On the first view, the team seems to have a chance of winning of only 50%. However, with a smart strategy, the chance of winning is almost 100%. More precisely, if the number of players has the form $n = 2^r - 1$, then the chance of winning is $n/(n+1)$.

The smart strategy is defined as follows. Let $\boldsymbol{y} = (y_0, \ldots, y_{n-1})$ be a vector representing the colors of the n hats. Let C be the $(2^r-1, 2^r-r-1, 3)_2$ Hamming code. By viewing his teammates, the i-th player knows the two vectors

$$\boldsymbol{a}_i = (y_0, \ldots, y_{i-1}, 0, y_{i+1}, \ldots, y_{n-1}),$$
$$\boldsymbol{b}_i = (y_0, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_{n-1})$$

Auf den ersten Blick beträgt die Gewinnchance des Teams nur 50%, da die Kenntnis der Hutfarben der Mitspieler keine Information über die eigene Hutfarbe vermittelt

… aber mit der richtigen Strategie und Hamming-Codes kann das Team fast immer gewinnen
… und das ist vollkommen gegen die Anschauung

Folgerung: man darf sich von der Anschauung inspirieren lassen aber am Ende zählt nur der exakte mathematische Beweis!

and guesses the color of his own hat as follows:

$$\begin{aligned} a_i \notin \mathcal{C} \text{ and } b_i \notin \mathcal{C} &\Rightarrow \text{neutral} \\ a_i \in \mathcal{C} \text{ and } b_i \notin \mathcal{C} &\Rightarrow \text{guess 1} \\ a_i \notin \mathcal{C} \text{ and } b_i \in \mathcal{C} &\Rightarrow \text{guess 0} \end{aligned}$$

The fourth case $a_i \in \mathcal{C}$ and $b_i \in \mathcal{C}$ is not possible since $d_H(a_i, b_i) = 1$, however, $\mathcal{C}$ has a minimum Hamming distance of 3.

Now we compute the chance of winning. Two cases have to be distinguished. Firstly, if $y \in \mathcal{C}$, then all players guess incorrectly and the team loses. Secondly, we consider the case of $y \notin \mathcal{C}$. Since $\mathcal{C}$ is perfect, there exists exactly one $c \in \mathcal{C}$ with $d_H(y, c) = 1$. Let l be the position where the two vectors differ.

The l-th player guesses as follows: if $a_l \in \mathcal{C}$, then he guesses 1 and $a_l \neq y$ since $y \notin \mathcal{C}$. Hence, $a_l = c \in \mathcal{C}$ and $b_l = y$ and so his guess is correct. The same arguments also show a correct guess in case of $b_l \in \mathcal{C}$. All other s-th players with $s \neq l$ remain neutral since $a_s \notin \mathcal{C}$ and $b_s \notin \mathcal{C}$.

In summary, in case of $y \notin \mathcal{C}$ one player guesses correctly and all other players remain neutral. Hence

$$P(y \notin \mathcal{C}) = \frac{2^n - 2^k}{2^n} = 1 - 2^{-(n-k)} = 1 - 2^{-r} = 1 - (n+1)^{-1} = \frac{n}{n+1}$$

is the teams's winning chance.

Strategie des Teams

Der Beweis der Gewinnchance ist eher öde und bietet keine weiteren Erkenntnisse.

Von geringer theoretischer aber großer praktischer Bedeutung sind folgende Modifikationen, die zu Codes mit geänderten Parametern führen:

Definition 4.6. *Ein* $(n, k, d_{\min})_q$*-Code kann wie folgt zu einem* $(n', k', d'_{\min})_q$*-Code verändert werden:*

(1) Beim Expandieren *(extending) werden zusätzliche Prüfbits angehängt:*
$n' > n, \quad k' = k, \quad R' < R, \quad d'_{\min} \geq d_{\min}.$

(2) Beim Punktieren *(puncturing) werden Prüfbits unterdrückt:*
$n' < n, \quad k' = k, \quad R' > R, \quad d'_{\min} \leq d_{\min}.$

(3) Beim Verlängern *(lengthening) werden zusätzliche Infobits angehängt:*
$n' > n, \quad k' > k, \quad n' - k' = n - k, \quad R' > R, \quad d'_{\min} \leq d_{\min}.$

(4) Beim Verkürzen *(shortening) werden Infobits unterdrückt:*
$n' < n, \quad k' < k, \quad n' - k' = n - k, \quad R' < R, \quad d'_{\min} \geq d_{\min}.$

Offensichtlich sind die Modifikationen 1 und 2 sowie 3 und 4 jeweils invers zueinander.

Am wichtigsten sind Punktieren und Verkürzen.

Diese Modifikationen haben keinen tiefsinnigen codierungstheoretischen Hintergrund, sondern sind einfach praktisch erforderlich wenn die Kanalcodierung auch die (wenig vornehme) Aufgabe hat, eine Anpassung zwischen Infobitrate r_b und Codebitrate r_c zu vermitteln.