9 Decodierung linearer Codes

Zur Erinnerung:

- Codierung mit Generatormatrix $\boldsymbol{G}$:

$$\boldsymbol{x} = \boldsymbol{a}\boldsymbol{G} \tag{9.1}$$

- Überprüfung auf Fehler mit Kontrollmatrix $\boldsymbol{H}$:

$$\boldsymbol{y}\boldsymbol{H}^T = \boldsymbol{0} \quad \Leftrightarrow \quad \boldsymbol{y} \in C \tag{9.2}$$

Wie korrigiert man Fehler, d.h. wie decodiert man, wenn $\boldsymbol{y}\boldsymbol{H}^T \neq \boldsymbol{0}$?

9.1 Syndrombasierte Decodierung

Definition 9.1 (Syndrom)

Das Ergebnis

$$\boldsymbol{s} = \boldsymbol{y}\boldsymbol{H}^T \tag{9.3}$$

der Multiplikation des Empfangsvektors $\boldsymbol{y}$ mit der Kontrollmatrix $\boldsymbol{H}$ wird Syndrom genannt.

Da

$$\begin{aligned} \boldsymbol{s} &= \boldsymbol{y}\boldsymbol{H}^T = (\boldsymbol{x}+\boldsymbol{e})\boldsymbol{H}^T && (9.4)\\ &= \underbrace{\boldsymbol{x}\boldsymbol{H}^T}_{\boldsymbol{0}} + \boldsymbol{e}\boldsymbol{H}^T && (9.5)\\ &= \boldsymbol{e}\boldsymbol{H}^T && (9.6) \end{aligned}$$

führt jedes Fehlermuster $\boldsymbol{e}$ auf ein von $\boldsymbol{x}$ unabhängiges Syndrom $\boldsymbol{s}$.

Beispiel 9.1

Der Wiederholungscode $\boldsymbol{G} = \begin{pmatrix}1 & 1 & 1\end{pmatrix}$ hat die Kontrollmatrix

$$\boldsymbol{H}^T = \begin{pmatrix}1 & 1 & 0\\ 1 & 0 & 1\end{pmatrix}.$$

Es gilt:

$$\begin{aligned}
(000+000)\boldsymbol{H}^T &= (111+000)\boldsymbol{H}^T = 00\\
(000+001)\boldsymbol{H}^T &= (111+001)\boldsymbol{H}^T = 01\\
(000+010)\boldsymbol{H}^T &= (111+010)\boldsymbol{H}^T = 10\\
(000+011)\boldsymbol{H}^T &= (111+011)\boldsymbol{H}^T = 11\\
(000+100)\boldsymbol{H}^T &= (111+100)\boldsymbol{H}^T = 11\\
(000+101)\boldsymbol{H}^T &= (111+101)\boldsymbol{H}^T = 10\\
(000+110)\boldsymbol{H}^T &= (111+110)\boldsymbol{H}^T = 01\\
(000+111)\boldsymbol{H}^T &= (111+111)\boldsymbol{H}^T = 00
\end{aligned}$$

Zur syndrombasierten Fehlerkorrektur wird zu jedem möglichen Syndrom das Fehlermuster mit der gerinsten Zahl an Fehlern tabelliert und zur Fehlerkorrektur benutzt.

Beispiel 9.2

Für den Wiederholungscode sähe die Tabelle so aus:

$\boldsymbol{s}$	$\boldsymbol{e}$
00	000
01	001
10	010
11	100

Wird $\boldsymbol{x} = 000$ gesendet, aber $\boldsymbol{y} = 010$ empfangen, so ist $\boldsymbol{s} = 010\boldsymbol{H}^T = 10$, also $\boldsymbol{e} = 010$, und der Empfänger kann $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{e} = 000$ rekonstruieren.
Wird hingegen $\boldsymbol{y} = 110$ empfangen, so ist $\boldsymbol{s} = 110\boldsymbol{H}^T = 01$, also $\boldsymbol{e} = 001$, und der Empfänger würde (fälschlicherweise) $\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{e} = 111$ rekonstruieren, da dieser Code nur einen Fehler korrigieren kann.

Beispiel 9.3

Der systematische Hamming-Code mit $k = n - l = 3$ Prüfbits hat die Kontrollmatrix

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

Durch systematisches Einsetzen von Fehlermustern mit steigender Anzahl von gestörten Stellen, kann dazu die Syndromtabelle gewonnen werden:

$\boldsymbol{s}$	$\boldsymbol{e}$
000	0000000
001	0000001
010	0000010
100	0000100
011	0001000
101	0010000
110	0100000
111	1000000

Ist auf diese Weise zu jedem Syndrom das Fehlermuster mit der geringsten Anzahl an Fehlern gefunden, endet die Erzeugung der Tabelle.

Beispiel 9.4

Es soll die Syndromtabelle für den Code mit der Kontrollmatrix

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

erzeugt werden:

$\boldsymbol{s}$	$\boldsymbol{e}$
000	00000
001	00001
010	00010
100	00100
101	01000
110	10000
011	00011
111	10001

9.2 Decodierung mittels Belief Propagation

- Die syndrombasierte Decodierung erfordert eine Syndromtabelle mit 2^k Einträgen, wodurch die praktisch möglich Anzahl k an Prüfbits stark eingeschränkt wird.
- Für geeignete Kontrollmatrizen kann eine Decodierung effizienter mit dem iterativen Verfahren der *Belief Propagation* erfolgen.
- Dazu wird der Tanner-Graph betrachtet.

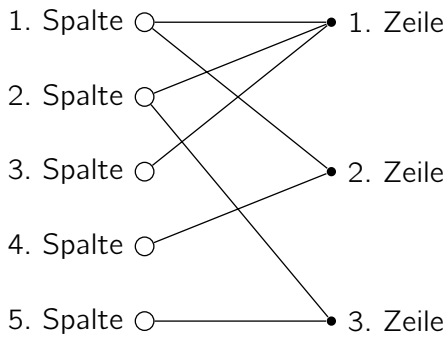
Der Tanner-Graph ist die Darstellung einer Kontrollmatrix als bipartiter Graph. Die Zeilen und Spalten bilden jeweils eine der beiden Knotenteilmengen. Zwei Knoten sind verbunden, wenn die Kontrollmatrix an der entsprechenden Stelle eine 1 enthält.

Beispiel 9.5

Der Tanner-Graph zu der Kontrollmatrix

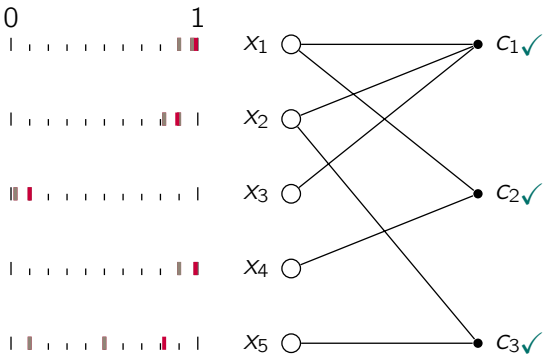
$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

sieht wie folgt aus:



Beispiel 9.6 (Prinzipieller Ablauf der Decodierung mit Belief Propagation)

Bei einer Codierung mit dem Code aus Beispiel 9.5 wurde das Wort 11010 empfangen und soll decodiert werden (das dichteste Codewort ist 11011).

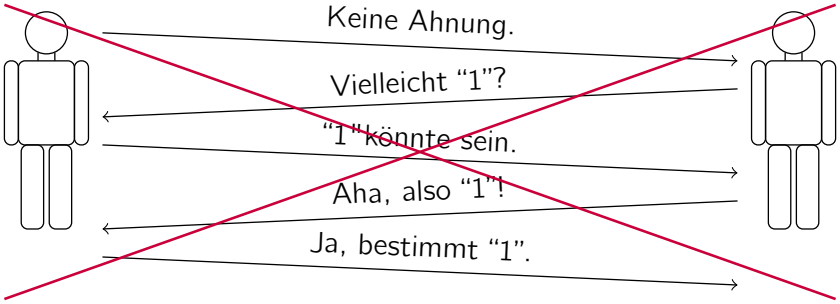


$\Rightarrow$ Erfolgreiche Decodierung zu 11011

Es werden abwechselnd Informationen von den Variablenknoten (entsprechend den empfangenen Bits) zu den Prüfknoten und wieder zurück geschickt.

1. Von den Variablenknoten x_i werden die Wahrscheinlichkeitsverteilungen $p_{X|Y}(x_i|y_i)$, die am Kanalausgang vorliegen, an die Prüfknoten c_j geschickt.
   Liefert z.B. ein symmetrisch gestörter Binärkanal mit Fehlerwahrscheinlichkeit p für y_i eine 1, so ist unter Annahme gleichverteilter Sendedaten $p_{X|Y}(0|1) = p$, $p_{X|Y}(1|1) = 1 - p$.
   Für den binären Fall ist die Wahrscheinlichkeitsverteilung durch nur eine Zahl beschrieben; im Folgenden sei $p_i = p_{X|Y}(1|y_i)$.
2. Von den Prüfknoten c_j werden die Wahrscheinlichkeiten r_{ji} an die Variablenknoten x_i geschickt, mit denen die jeweiligen Bedingungen c_j für $x_i = 1$ erfüllt werden, die sich aus den aktuellen Wahrscheinlichkeitsverteilungen aller jeweils anderen mit c_j verknüpften Variablenknoten ergeben.
   Ist beispielsweise c_1 nur mit x_1 und x_2 verknüpft, so ist $r_{1,1} = p_2$ und $r_{1,2} = p_1$.
3. Von den Variablenknoten x_i werden die mittels der r_{ji} aktualisierten Wahrscheinlichkeiten q_{ji} an die Prüfknoten c_j geschickt. Dabei werden jeweils für q_{ji} alle $r_{j'i}$ *außer* r_{ji} berücksichtigt, um eine "Selbstbekräftigung" bei c_j zu verhindern.
4. Weiter bei 2.

Grundsätzlich ist es bei der Belief Propagation wichtig, dass Nachrichten, die von einem Knoten A an einen Knoten B geschickt werden, nicht auf Nachrichten beruhen, die A zuvor von B erhalten hat.

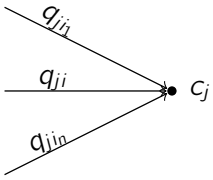


Sichere Überzeugung aus dem nichts!

9.2.1 Berechnung an den Prüfknoten

Die Variablenknoten x_i, die mit dem Prüfknoten c_j verbunden sind, haben diesem die Wahrscheinlichkeit q_{ji} gemeldet, dass $x_i = 1$.
Die an einen bestimmten Variablenknoten x_i zurückzusendende Nachricht ist die Wahrscheinlichkeit

$$r_{ji} = P(c_j \text{ wird mit } x_i = 1 \text{ erfüllt}|\text{aktuelle } q_{ji'} \text{ außer } q_{ji}). \tag{9.7}$$



Die Bedingung c_j ist erfüllt, wenn $\sum_{i' \in X_j} x_{i'} = 0$, unter Annahme von $x_i = 1$ also wenn $\sum_{i' \in X_j \setminus i} x_{i'} = 1$, wobei X_j die Menge der Indizes aller mit c_j verknüpften $x_{i'}$ bezeichnet.
Also

$$r_{ji} = P\left(\sum_{i' \in X_j \setminus i} x_{i'} = 1\right) \tag{9.8}$$

unter Verwendung der aktuellen q_{ij}.

Satz 9.1

Seien $x_i \in \{0, 1\}$, $i = 1, \ldots, N$ Zufallsvariablen mit $P(x_i = 1) = p_i$, so ist

$$P\left(\sum_{i=1}^{N} x_i \mod 2 = 1\right) = \frac{1}{2}\left(1 - \prod_{i=1}^{N}(1 - 2p_i)\right). \tag{9.9}$$

Beweis über vollständige Induktion.
Für $N = 1$ gilt offenbar

$$P\left(\sum_{i=1}^{N} x_i \mod 2 = 1\right) = p_1 = \frac{1}{2}\big(1 - (1 - 2p_1)\big). \tag{9.10}$$

Es bleibt zu zeigen, dass wenn

$$P_{N-1} = P\left(\sum_{i=1}^{N-1} x_i \mod 2 = 1\right) = \frac{1}{2}\left(1 - \prod_{i=1}^{N-1}(1 - 2p_i)\right) \tag{9.11}$$

(Induktionsvoraussetzung) gilt, auch

$$P_N = P\left(\sum_{i=1}^{N} x_i \mod 2 = 1\right) = \frac{1}{2}\left(1 - \prod_{i=1}^{N}(1 - 2p_i)\right) \tag{9.12}$$

gilt. □

Forsetzung des Beweises.

Es gilt

$$\begin{aligned} P_N &= P_{N-1} \cdot (1 - p_N) + (1 - P_{N-1}) \cdot p_N = P_{N-1} + p_N - 2P_{N-1}p_N \\ &= \frac{1}{2}(2P_{N-1} + 2p_N - 4P_{N-1}p_N) = \frac{1}{2}\big(1 - (1 - 2P_{N-1} - 2p_N + 4P_{N-1}p_N)\big) \\ &= \frac{1}{2}\big(1 - (1 - 2P_{N-1}) \cdot (1 - 2p_N)\big) \end{aligned} \tag{9.13}$$

Nach Induktionsannahme ist

$$P_{N-1} = \frac{1}{2}\left(1 - \prod_{i=1}^{N-1}(1 - 2p_i)\right), \tag{9.14}$$

Einsetzen liefert

$$\begin{aligned} P_N &= \frac{1}{2}\left(1 - \left(1 - \left(1 - \prod_{i=1}^{N-1}(1 - 2p_i)\right)\right) \cdot (1 - 2p_N)\right) \\ &= \frac{1}{2}\left(1 - \left(\prod_{i=1}^{N-1}(1 - 2p_i)\right) \cdot (1 - 2p_N)\right) = \frac{1}{2}\left(1 - \prod_{i=1}^{N}(1 - 2p_i)\right). \end{aligned} \tag{9.15}$$

□

Für jeden Prüfknoten c_j ist also für jeden verbundenen Variablenknoten x_i die Berechnung

$$r_{ji} = \frac{1}{2}\left(1 - \prod_{i' \in X_j \setminus i}(1 - 2q_{ji'})\right) \tag{9.16}$$

durchzuführen.

Beispiel 9.7

Sei c_1 mit x_1, x_2 und x_3 verbunden und $q_{1,1} = q_{1,2} = 0{,}9$, $q_{1,3} = 0{,}1$.

$$r_{1,1} = \frac{1}{2}\left(1 - (1 - 2q_{1,2}) \cdot (1 - 2q_{1,3})\right) = \frac{1}{2}\left(1 - (-0{,}8) \cdot 0{,}8\right) = 0{,}82 \tag{9.17}$$

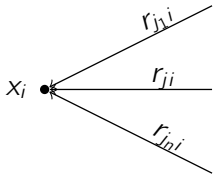
$$r_{1,2} = \frac{1}{2}\left(1 - (1 - 2q_{1,1}) \cdot (1 - 2q_{1,3})\right) = \frac{1}{2}\left(1 - (-0{,}8) \cdot 0{,}8\right) = 0{,}82 \tag{9.18}$$

$$r_{1,3} = \frac{1}{2}\left(1 - (1 - 2q_{1,1}) \cdot (1 - 2q_{1,2})\right) = \frac{1}{2}\left(1 - (-0{,}8) \cdot (-0{,}8)\right) = 0{,}18 \tag{9.19}$$

9.2.2 Berechnung an den Variablenknoten

Die Prüfknoten c_j, die mit dem Variablenknoten x_i verbunden sind, haben diesem die Wahrscheinlichkeit r_{ji} gemeldet, dass c_j erfüllt ist, wenn $x_i = 1$.
Die an einen bestimmten Prüfknoten c_j zurückzusendende Nachricht ist die Wahrscheinlichkeit

$$q_{ji} = P(x_i = 1 | y_i, \text{aktuelle } r_{j'i} \text{ außer } r_{ji}). \tag{9.20}$$



Sei C_i die Menge der Indizes aller mit x_i verknüpften c_j, so lässt sich schreiben

$$q_{ji} = P(x_i = 1 | y_i, c_{j'} \text{ für } j' \in C_i \backslash j). \tag{9.21}$$

Mit dem Satz von Bayes lässt sich umformen zu

$$q_{ji} = \frac{P(x_i = 1, y_i, c_{j'} \text{ für } j' \in C_i \backslash j)}{P(y_i, c_{j'} \text{ für } j' \in C_i \backslash j)} \tag{9.22}$$

$$q_{ji} = \frac{P(x_i = 1, y_i, c_{j'} \text{ für } j' \in C_i \backslash j)}{P(y_i, c_{j'} \text{ für } j' \in C_i \backslash j)} \tag{9.23}$$

$$= \frac{P(c_{j'} \text{ für } j' \in C_i \backslash j' | y_i, x_i = 1) \cdot P(x_i = 1 | y_i) \cdot P(y_i)}{P(y_i, c_{j'} \text{ für } j' \in C_i \backslash j')} \tag{9.24}$$

$$= P(c_{j'} \text{ für } j' \in C_i \backslash j | \cancel{y_i}, x_i = 1) \cdot P(x_i = 1 | y_i) \cdot \underbrace{\frac{P(y_i)}{P(y_i, c_{j'} \text{ für } j' \in C_i \backslash j)}}_{\alpha} \tag{9.25}$$

$$= \alpha \cdot P(x_i = 1 | y_i) \cdot \prod_{j' \in C_i \backslash j} r_{j'i} \tag{9.26}$$

Entsprechende Überlegungen führen für $x_{i'} = 0$ auf

$$\bar{q}_{ji} = \alpha \cdot P(x_i = 0 | y_i) \cdot \prod_{j' \in C_i \backslash j} (1 - r_{j'i}) \tag{9.27}$$

und mit $q_{ji} + \bar{q}_{ji} = 1$ lässt sich schließlich α bestimmen.

Beispiel 9.8

Sei x_1 mit c_1 und c_2 verbunden und $r_{1,1} = 0{,}82$, $r_{2,1} = 0{,}9$ sowie $p_{X|Y}(1, y_1) = 0{,}9$.

$$
\begin{aligned}
q_{1,1}/\alpha &= P(x_1 = 1|y_1) \cdot r_{2,1} = 0{,}9 \cdot 0{,}9 = 0{,}81 \\
\bar{q}_{1,1}/\alpha &= P(x_1 = 0|y_1) \cdot (1 - r_{2,1}) = 0{,}1 \cdot 0{,}1 = 0{,}01 \\
q_{1,1} &= 0{,}81/0{,}82 = 0{,}9878 \\
q_{2,1}/\alpha &= P(x_1 = 1|y_1) \cdot r_{1,1} = 0{,}9 \cdot 0{,}82 = 0{,}738 \\
\bar{q}_{2,1}/\alpha &= P(x_1 = 0|y_1) \cdot (1 - r_{1,1}) = 0{,}1 \cdot 0{,}18 = 0{,}018 \\
q_{2,1} &= 0{,}738/0{,}756 = 0{,}9762
\end{aligned}
$$

Zusätzlich lässt sich die Posteriori-Wahrscheinlichkeit

$$\hat{p}_i = \alpha \cdot P(x_i = 1|y_i) \cdot \prod_{j \in C_i} r_{ji} \tag{9.28}$$

ermitteln, mit der $x_i = 1$ unter Einbeziehung aller bis dahin verfügbaren Informationen an x_i.

Beispiel 9.9

Für obiges Beispiel:

$$
\begin{aligned}
\hat{p}_1/\alpha &= P(x_1 = 1|y_1) \cdot r_{1,1} \cdot r_{2,1} = 0{,}9 \cdot 0{,}82 \cdot 0{,}9 = 0{,}6642 \\
\bar{\hat{p}}_1/\alpha &= P(x_1 = 0|y_1) \cdot (1 - r_{1,1}) \cdot (1 - r_{2,1}) = 0{,}1 \cdot 0{,}18 \cdot 0{,}1 = 0{,}0018 \\
\hat{p}_1 &= 0{,}6642/0{,}666 = 0{,}9973
\end{aligned}
$$

9.2.3 Zusammenfassung des Decodier-Algorithmus

1. Initialisiere $q_{ji} = p_{X|Y}(1|y_i)$.
2. Berechne die Nachrichten

$$r_{ji} = \frac{1}{2}\left(1 - \prod_{i' \in X_j \setminus i} (1 - 2q_{ji'})\right) \tag{9.29}$$

   von allen Prüfknoten c_j an alle verbundenen Variablenknoten x_i.
3. Berechne mittels

$$q_{ji}/\alpha = P(x_i = 1|y_i) \cdot \prod_{j' \in C_i \setminus j} r_{j'i} \tag{9.30}$$

$$\bar{q}_{ji}/\alpha = P(x_i = 0|y_i) \cdot \prod_{j' \in C_i \setminus j} (1 - r_{j'i}) \tag{9.31}$$

   die Nachrichten

$$q_{ji} = \frac{q_{ji}/\alpha}{q_{ji}/\alpha + \bar{q}_{ji}/\alpha} \tag{9.32}$$

   von allen Variablenknoten x_i an alle verbundenen Prüfknoten c_j.
4. Wiederhole ab 2.

Der Algorithmus kann abgebrochen werden

- nach einer vorgegebenen Anzahl von Iterationen,
- oder wenn eine Bitentscheidung anhand der Posteriori-Wahrscheinlichkeiten (siehe unten) auf eine gültiges Codewort führt.

Das Ergebnis der Decodierung sind die Posteriori-Wahrscheinlichkeiten

$$\hat{p}_i = \alpha \cdot P(x_i = 1|y_i) \cdot \prod_{j \in C_i} r_{ji} \tag{9.33}$$

bzw. die daraus entschiedenen Bits

$$\hat{x}_i = \begin{cases} 1 & \text{falls } \hat{p}_i \geq 0{,}5 \\ 0 & \text{sonst.} \end{cases} \tag{9.34}$$

Man kann zeigen, dass für einen zyklenfreien Tanner-Graphen nach endlich vielen Iterationen $\hat{p}_i = P_{X_i|\boldsymbol{Y}}(1|\boldsymbol{y})$ gilt, also die Wahrscheinlichkeit eines einzelnen Bits in Abhängigkeit des gesamten Empfangswortes und unter Berücksichtigung des Codes bestimmt wird.
Enthält der Graph Zyklen, wird dieses Ergebnis nur näherungsweise erreicht.

9.3 LDPC-Codes

- Bei der Decodierung mittels Belief Propagation hängt der Rechnenaufwand nicht von der Größe der Prüfmatrix, sondern von der Anzahl der Kanten im Tanner-Graph, also der Anzahl der Einsen in der Prüfmatrix ab.

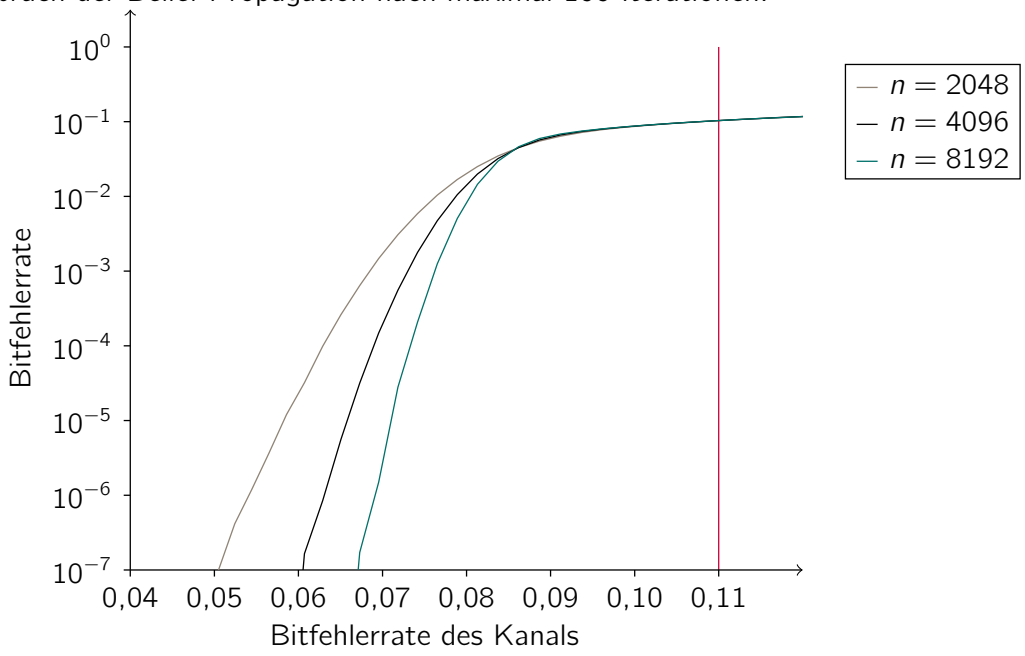
$\Rightarrow$ Dünn besetzte Prüfmatrix verwenden!

- Eine dünn besetzte Matrix hilft auch, Zyklen im Tanner-Graphen zu reduzieren, ist also besonders geeignet für eine Decodierung mittels Belief Propagation.

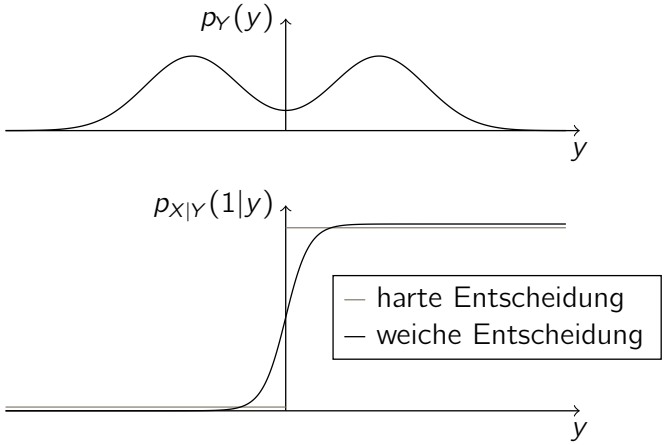
Codes, die auf einer dünn besetzten Prüfmatrix beruhen, werden Low Density Parity Check (LDPC) Codes genannt.
Typische Erzeugung: Zufällige Prüfmatrizen mit vorgegenem Verhältnis von Einsen und Nullen und weiteren Einschränkungen (Spaltengewichte, Zeilengewichte, ...).
Durch Umsortieren der Spalten lässt sich jede Prüfmatrix (mit unabhängigen Zeilen) in eine Form $\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H_1} & \boldsymbol{H_2} \end{pmatrix}$ mit invertierbarer Untermatrix $\boldsymbol{H_2}$ bringen. Damit hat $\boldsymbol{H_2}^{-1}\boldsymbol{H} = \begin{pmatrix} \boldsymbol{H_2}^{-1}\boldsymbol{H_1} & \boldsymbol{I} \end{pmatrix}$ denselben Nullraum (beschreibt also den gleichen Code) und $\boldsymbol{G} = \begin{pmatrix} \boldsymbol{I} & (\boldsymbol{H_2}^{-1}\boldsymbol{H_1})^T \end{pmatrix}$ ist eine systematische Generatormatrix zu $\boldsymbol{H}$.

Beispiel 9.10 (LDPC-Code für einen symmetrisch gestörten Binärkanal)

Verwendete Prüfmatrizen: Coderate $R = 1/2$, drei Einsen pro Spalte, sechs Einsen pro Zeile, maximal eine gemeinsame Eins zwischen zwei Spalten.
Abbruch der Belief Propagation nach maximal 100 Iterationen.

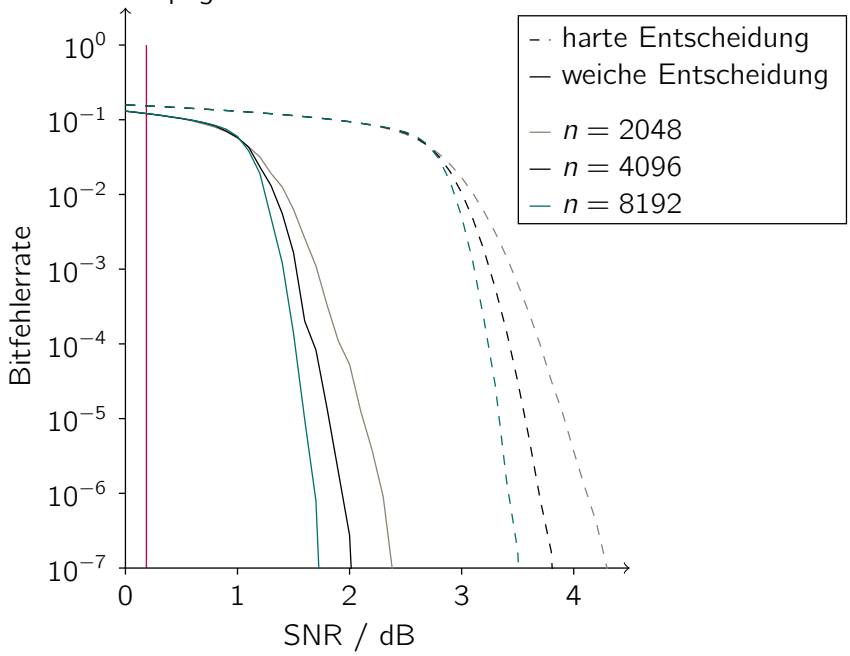


Bei der Übertragung über analoge Kanäle (z.B. AWGN) lässt sich mehr Information als die harte Bitentscheidung (Rückführung auf einen Binärkanal) gewinnen.



Beispiel 9.11 (LDPC-Code für einen AWGN-Kanal)

Verwendete Prüfmatrizen: Coderate $R = 1/2$, drei Einsen pro Spalte, sechs Einsen pro Zeile, maximal eine gemeinsame Eins zwischen zwei Spalten.
Abbruch der Belief Propagation nach maximal 100 Iterationen.



9.4 Zusammenfassung

- Die syndrombasierte Decodierung basiert auf einer Tabelle, die zu jeden möglichen Syndrom das Fehlermuster mit der geringsten Anzahl an Fehlern enthält.
- Da die Größe der Syndromtabelle exponentiell mit der Anzahl k der Prüfbits steigt, wird die syndrombasierte Decodierung schnell inpraktikabel.
- Die Belief Propagation ist ein iteratives Verfahren, mit dem aus den Wahrscheinlichkeiten $p_{X|Y}(x_i|y_i)$ am Kanalausgang die Wahrscheinlichkeiten $p_{X_i|\boldsymbol{Y}}(x_i|\boldsymbol{y})$ unter Berücksichtigung des Codes berechnet bzw. angenähert werden können.
- Der Rechnenaufwand der Belief Propagation hängt im Wesentlichen linear von der Anzahl der Einsen in der Prüfmatrix ab.
- Die LDPC-Codes nutzen daher große, dünn besetzte Prüfmatrizen, um eine Codierung nahe an der Shannon-Grenze zu ermöglichen.
- Durch eine "weiche" Entscheidung am Ausgang eines analogen Kanals ist eine weitere deutliche Verbesserung der Leistungsfähigkeit möglich.