Benedikt Wimmer

universität innsbruck

R-Paket für Kanalkodierung mit Blockkodes

Gegründet im Jahr 1669, ist die Universität Innsbruck heute mit mehr als 28.000 Studierenden und über 4.000 Mitarbeitenden die größte und wichtigste Forschungs- und Bildungseinrichtung in Westösterreich. **Alle weiteren Informationen finden Sie im Internet unter: www.uibk.ac.at.**

Gliederung


Nachricht
N?ch?icht

Motivation und Ziel

- Kodierer sind hauptsächlich in der Hardware realisiert.

Motivation und Ziel

- Kodierer sind hauptsächlich in der Hardware realisiert.
- Schlecht nachzuvollziehen, Lehre bisher nur theoretisch.

Motivation und Ziel

- Kodierer sind hauptsächlich in der Hardware realisiert.
- Schlecht nachzuvollziehen, Lehre bisher nur theoretisch.
- **Ziel:** R-Paket zur Simulation und Visualisierung der Kanalkodierung.

Grundlagen

- Kodierung als Abbildung

$$D^k \rightarrow K^n$$

Grundlagen

- Kodierung als Abbildung

$$D^k \rightarrow K^n$$

- Koderate R

$$R = \frac{k}{n}$$

Grundlagen

- Kodierung als Abbildung
$$D^k \to K^n$$
- Koderate R
$$R = \frac{k}{n}$$
- Hamming Distanz d für $x, y \in K$
$$d(x, y) := \mid \{i \in \{1, ..., n\} \mid x_i \neq y_i\} \mid$$

Grundlagen

- Kodierung als Abbildung

$$D^k \to K^n$$

- Koderate R

$$R = \frac{k}{n}$$

- Hamming Distanz d für $x, y \in K$

$$d(x, y) := \mid \{i \in \{1, ..., n\} \mid x_i \neq y_i\} \mid$$

- Beispiel

$$d(1011, 1010) = 1$$

Grundlagen - Fehlermodell

- Additive White Gaussian Noise (AWGN)

$$x_i = y_i + Z_i \sim \mathcal{N}(0, \sigma^2)$$

Grundlagen - Fehlermodell

- Additive White Gaussian Noise (AWGN)

$$x_i = y_i + Z_i \sim \mathcal{N}(0, \sigma^2)$$

- Signal-Rausch-Verhältnis

$$SNR = 10 \log_{10} \left(\frac{Signalleistung\ S}{Rauschleistung\ N} \right) dB$$

Grundlagen - Fehlermodell

- Additive White Gaussian Noise (AWGN)

$$x_i = y_i + Z_i \sim \mathcal{N}(0, \sigma^2)$$

- Signal-Rausch-Verhältnis

$$SNR = 10 \log_{10} \left(\frac{Signalleistung\ S}{Rauschleistung\ N} \right) dB$$

- Shannon Grenze

$$C_{max} \leq B \log_2 \left(1 + \frac{S}{N} \right)$$

Blockkodes - Allgemeines

1. Nachricht in Blöcke aufteilen

Blockkodes - Allgemeines

1. Nachricht in Blöcke aufteilen
2. Jeden Block kodieren / dekodieren

Blockkodes - Allgemeines

1. Nachricht in Blöcke aufteilen
2. Jeden Block kodieren / dekodieren
3. Blöcke zur kodierten / dekodierten Nachricht zusammensetzen

Blockkodes - Buchstabieralphabet

Daten	*Kodierung*
A	`Anton`
B	`Berta`
C	`Cäsar`
.	.
.	.
.	.
Z	`Zeppelin`

Hamming-Kodes

- Hamming-(n,k)-Kode mit $n = 2^m - 1$ und $k = 2^m - m - 1$

Quelle: [1]

Hamming-Kodes

- Hamming-(n,k)-Kode mit $n = 2^m - 1$ und $k = 2^m - m - 1$
- Hamming-Distanz $d = 3$, kann 1 Fehler korrigieren.

Quelle: [1]

Hamming-Kodes

- Hamming-(n,k)-Kode mit $n = 2^m - 1$ und $k = 2^m - m - 1$
- Hamming-Distanz $d = 3$, kann 1 Fehler korrigieren.
- Generatormatrix $G = (I_k \mid A)$

Quelle: [1]

Hamming-Kodes

- Hamming-(n,k)-Kode mit $n = 2^m - 1$ und $k = 2^m - m - 1$
- Hamming-Distanz $d = 3$, kann 1 Fehler korrigieren.
- Generatormatrix $G = (I_k \mid A)$
- Kontrollmatrix $H = \left(A^T \mid I_m\right)$

Quelle: [1]

Hamming-Kodes - (7,4)

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

Hamming-Kodes - (7,4)

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

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

Hamming-Kodes - (7,4)

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

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

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

Hamming-Kodes - (7,4)

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

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

Hamming - (7,4) - Kodierung

$$\begin{pmatrix}1\\0\\1\\0\end{pmatrix}^T \cdot \begin{pmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&1&1&1\end{pmatrix} = \begin{pmatrix}1\\0\\1\\0\\1\\0\\1\end{pmatrix}^T$$

Hamming - (7,4) - Dekodierung

$$\begin{pmatrix}1\\0\\1\\0\\1\\0\\1\end{pmatrix}^T \cdot \begin{pmatrix}0 & 1 & 1 & 1 & 1 & 0 & 0\\1 & 0 & 1 & 1 & 0 & 1 & 0\\1 & 1 & 0 & 1 & 0 & 0 & 1\end{pmatrix} = \begin{pmatrix}0\\0\\0\end{pmatrix}^T$$

Kodewort: (1,0,1,0,1,0,1)
Datenwort: (1,0,1,0)

Hamming - (7,4) - Dekodierung

$$\begin{pmatrix}1\\1\\1\\0\\1\\0\\1\end{pmatrix}^T \cdot \begin{pmatrix}0 & 1 & 1 & 1 & 1 & 0 & 0\\1 & 0 & 1 & 1 & 0 & 1 & 0\\1 & 1 & 0 & 1 & 0 & 0 & 1\end{pmatrix} = \begin{pmatrix}1\\0\\1\end{pmatrix}^T$$

Kodewort: (1,1,1,0,1,0,1)
Fehlerindex: 2
Korrigiertes Kodewort: (1,0,1,0,1,0,1)
Datenwort: (1,0,1,0)

BCH-Kodes

- **B**ose-**C**haudhuri-**H**ocquenghem

BCH-Kodes

- **B**ose-**C**haudhuri-**H**ocquenghem
- **(n,k,d)**-Kode

BCH-Kodes

- **B**ose-**C**haudhuri-**H**ocquenghem
- **(n,k,d)**-Kode
- Korrigiert auf n-Bit Kodewörter bis zu $t = \frac{d}{2} - 1$ Fehler

BCH - Kodierung

- Nachrichten werden als Polynome mit Koeffizienten aus $GF(2)$ interpretiert.

BCH - Kodierung

- Nachrichten werden als Polynome mit Koeffizienten aus $GF(2)$ interpretiert.
- Generatorpolynom $g(x)$ kodiert Datenwort $dw(x)$:

$$kw(x) = dw(x)x^{n-k} - \left(dw(x)x^{n-k} \mod g(x)\right)$$

BCH - (15,5,7) - Kodierung

Input: (1,0,1,0,1)

Daten Polynom:

$$dw(x) = x^4 + x^2 + 1$$

BCH - (15,5,7) - Kodierung

Input: (1,0,1,0,1)

Daten Polynom:

$$dw(x) = x^4 + x^2 + 1$$

Generator Polynom:

$$g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x^1 + 1$$

$$dw^*(x) = dw(x) * x^{10} = x^{14} + x^{12} + x^{10}$$

BCH - (15,5,7) - Kodierung

Input: (1,0,1,0,1)

Daten Polynom:

$$dw(x) = x^4 + x^2 + 1$$

Generator Polynom:

$$g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x^1 + 1$$

$$dw^*(x) = dw(x) * x^{10} = x^{14} + x^{12} + x^{10}$$

Rest Polynom:

$$r(x) = dw^*(x) \mod g(x) = x^9 + x^6 + x^2 + x^1 + 1$$

BCH - (15,5,7) - Kodierung

Input: (1,0,1,0,1)
Daten Polynom:

$$dw(x) = x^4 + x^2 + 1$$

Generator Polynom:

$$g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x^1 + 1$$

$$dw^*(x) = dw(x) * x^{10} = x^{14} + x^{12} + x^{10}$$

Rest Polynom:

$$r(x) = dw^*(x) \mod g(x) = x^9 + x^6 + x^2 + x^1 + 1$$

Kode Polynom:

$$kw(x) = c(r(x)|dw(x)) = x^{14} + x^{12} + x^{10} + x^9 + x^6 + x^2 + x^1 + 1$$

Output: (1,1,1,0,0,0,1,0,0,1,1,0,1,0,1)

BCH - Generatorpolynom

Körpererweiterung $GF(2^m)$ konstruieren. Beispiel für $m = 3$:

BCH - Generatorpolynom

Körpererweiterung $GF(2^m)$ konstruieren. Beispiel für $m = 3$:

α^i	Polynom	Vektor	Minimalpolynom
-	0	000	-
α^0	1	100	-
α^1	x	010	$x^3 + x + 1$
α^2	x^2	001	$x^3 + x + 1$
α^3	$x + 1$	110	$x^3 + x^2 + 1$
α^4	$x^2 + x$	011	$x^3 + x + 1$
α^5	$x^2 + x + 1$	111	$x^3 + x^2 + 1$
α^6	$x^2 + 1$	101	$x^3 + x^2 + 1$

BCH - Generatorpolynom

- Generatorpolynom für t-Fehler korrigierenden Kode hat Nullstellen

$$\alpha^1, ..., \alpha^{2t}$$

BCH - Generatorpolynom

- Generatorpolynom für t-Fehler korrigierenden Kode hat Nullstellen

$$\alpha^1, ..., \alpha^{2t}$$

- Für Minimalpolynome ϕ_i:

$$g(x) = LCM(\phi_1(x), \phi_2(x), ..., \phi_{2t}(x))$$

BCH - Dekodierung

1. Berechnen der Syndrome $S_i = kw(\alpha^i) \mid 1 \leq i \leq 2t$.

Quelle: [2]

BCH - Dekodierung

1. Berechnen der Syndrome $S_i = kw(\alpha^i) \mid 1 \leq i \leq 2t$.
2. Bestimmen des Fehlerstellenpolynoms $\sigma(x)$.
   (Berlekamp-Massey-Algorithmus)

Quelle: [2]

BCH - Dekodierung

1. Berechnen der Syndrome $S_i = kw(\alpha^i) \mid 1 \leq i \leq 2t$.
2. Bestimmen des Fehlerstellenpolynoms $\sigma(x)$.
   (Berlekamp-Massey-Algorithmus)
3. Ermitteln der Inversen der Nullstellen von $\sigma(x)$, diese entsprechen den Fehlerindizes α^i. (Chien's Suche)

Quelle: [2]

BCH - Dekodierung

1. Berechnen der Syndrome $S_i = kw(\alpha^i) \mid 1 \leq i \leq 2t$.
2. Bestimmen des Fehlerstellenpolynoms $\sigma(x)$.
   (Berlekamp-Massey-Algorithmus)
3. Ermitteln der Inversen der Nullstellen von $\sigma(x)$, diese entsprechen den Fehlerindizes α^i. (Chien's Suche)
4. Korrigieren der Fehler im Kodewort an den Fehlerindizes.

Quelle: [2]

Verwendung von Blockkodes

- Hamming-Kodes fast nur zu Lehrzwecken.

Verwendung von Blockkodes

- Hamming-Kodes fast nur zu Lehrzwecken.
- Reed-Solomon-Kodes in der Datenspeicherung.

Verwendung von Blockkodes

- Hamming-Kodes fast nur zu Lehrzwecken.
- Reed-Solomon-Kodes in der Datenspeicherung.
- LDPC und Turbo-Kodes lösen Blockkodes ab.

Programmiersprache R

- Ursprünglich für statistische Zwecke entwickelt.

Quelle: [3]

Programmiersprache R

- Ursprünglich für statistische Zwecke entwickelt.
- Über 8000 Pakete auf den CRAN-Servern.

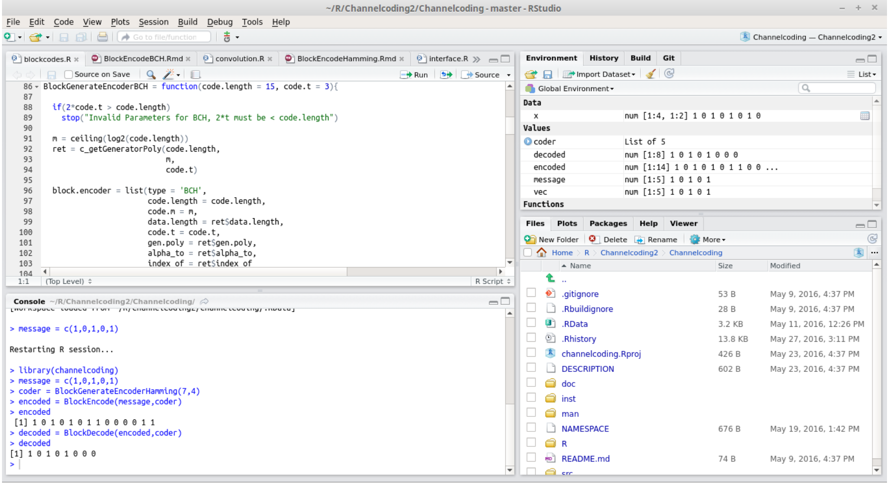
Quelle: [3]

Programmiersprache R

- Ursprünglich für statistische Zwecke entwickelt.
- Über 8000 Pakete auf den CRAN-Servern.
- Bereits in Lehrveranstaltungen verwendet.

Quelle: [3]

Entwicklungsumgebung RStudio



Literatur

- [1] Cary W. Huffman und Vera Pless. *Fundamentals of error-correcting codes.* Cambridge university press, 2010.
- [2] Robert H. Morelos-Zaragoza. *The art of error correcting coding.* John Wiley and Sons, 2006.
- [3] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria, 2016.