Zusammenfassung Information Theory and Coding

Markus Velm

Inhaltsverzeichnis

1. Einleitung

1.1. Informationstheorie

Bit: binary unit → Einheit für Information

bit: binary digit → bit als binäres Symbol

Informationgehalt

je unwahrscheinlicher ein Symbol x auftritt, desto mehr Information enhält es:

$$I(x) = ld\left(\frac{1}{P(x)}\right) = -ld(P(x))$$

P: Wahrscheinlichkeit eines Symbols
I: Informationsgehalt $[I] = Bit$

Entropie

gemittelter Informationsgehalt einer Quelle X:

$$H(X) = \sum_i P(x_i) \cdot I(x_i) = -\sum_i P(x_i) \cdot ld(P(x_i))$$

H: Entropie $[H] = Bit/Symbol$

Entscheidungsgehalt

Entropie wird maximal, wenn alle Symbole gleichwahrscheinlich sind → Entscheidungsgehalt

$$H_0 = ld(N)$$

H_0: Entscheidungsgehalt $[H_0] = Bit/Symbol$
N: Anzahl der Symbole eines Alphabets

Redundanz

$$R = H_0 - H$$

$$r = \frac{R}{H_0}$$

R: Redundanz $[R] = Bit/Symbol$
r: relative Redundanz

1.2. Quellcodierung

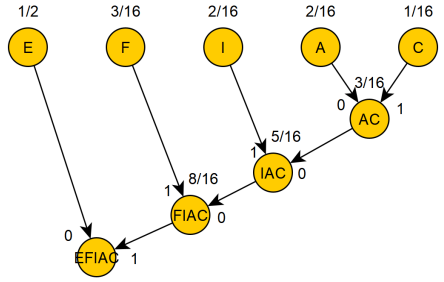
1.2.1. Huffman-Code

ist **Präfixcode:** ein Codewort ist niemals Anfang eines anderen Codewortes

Codebaum aufbauen:

1. Ordne die Symbole nach Auftrittswahrscheinlichkeit
2. Fasse Symbole mit niedrigster Wahrscheinlichkeit zu einem Symbol zusammen und addiere die Wahrscheinlichkeiten
3. Wiederhole bis nur ein Symbol übrig bleibt

Beschrifte die Pfade mit 1 und 0
→ Codewort ergibt sich, indem man von Wurzel bis zum Blatt geht



Hinweis: Beschriftung der 0; 1 theoretisch egal aber für Überprüfung mit Onlinerechnern sollte konsistent der Pfad mit der geringeren und höheren Wahrscheinlichkeit gleich beschriftet werden

1.2.2. Arithmetische Codierung

Codierung eines Wortes (oder Textes) durch Zahl

Endezeichen notwendig, da keine natürliche Terminierung des Codes

Codierung

Den Symbolen wird ein Interval zwischen 0..1 (oder beliebig, z.B. zwischen 0...1024)

Je größer die Wahrscheinlichkeit für Symbol, desto größer das Interval

Bsp.:

$P(A) = 0,5 \quad P(B) = 0,25 \quad P(C) = 0,25$

$$\overset{A}{|_0 - - - - -}\overset{B}{|_{0,5} - - -}\overset{C}{|_{0,75} - - -}|_1$$

Decodierung

empfangen wird eine Dezimalzahl r (oder eine ganze Zahl)

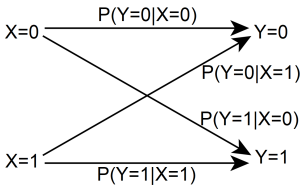
$$c_0 = r$$

$$c_n = \frac{c_{n-1} \cdot low}{high - low}$$

Decodierung bis Endezeichen kommt

1.3. Kanalmodell

1.3.1. Binärkanal



$P(Y|X)$: »Wahrscheinlichkeit für Y, wenn X gesendet wurde«(a-priori-Wahrscheinlichkeit)

$P(X|Y)$: »Wahrscheinlichkeit für X, wenn Y empfangen wurde«(a-posteriori-Wahrscheinlichkeit)

Unxymmetrisch: Fehlerwahrscheinlichkeit für »0« und »1« unterschiedlich

$$(P(Y|X)) = \begin{pmatrix} P(0|0) & P(0|1) \\ P(1|0) & P(1|1) \end{pmatrix}$$

Symmetrisch: Bitfehlerwahrscheinlichkeit P_e für »0« und »1« gleich

$$(P(Y|X)) = \begin{pmatrix} P(0|0) & P(0|1) \\ P(1|0) & P(1|1) \end{pmatrix} = \begin{pmatrix} 1-P_e & P_e \\ P_e & 1-P_e \end{pmatrix}$$

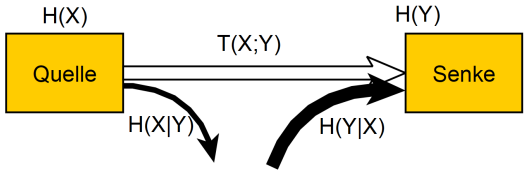
1.3.2. Bedingte Entropie

$$H(Y|X) = -\sum_j \sum_i P(x_i, y_i)\, ld(P(y_i|x_i)) \quad \textbf{Irrelevanz}$$

$$H(X|Y) = -\sum_j \sum_i P(x_i, y_i)\, ld(P(x_i|y_i)) \quad \textbf{Äquivokation}$$

$H(Y|X)$: **Irrelevanz (Fehlinformation)**, mittlere Unsicherheit über empfangene Symbole, wenn Sendesymbole bekannt
$H(X|Y)$: **Äquivokation (Informationsverlust)**, mittlere Unsicherheit über gesendete Symbole, wenn Empfangssymbole bekannt

1.3.3. Entropiemodell



für Empfänger ist $H(Y)$ (also der Ausgang des Kanals) eine neue Quelle, allerdings mit zusätzlicher Fehlinformation

Die Transinformation ist dann die Entropie des Kanals minus der Fehlinformation

$$T(X;Y) = H(Y) - \underbrace{H(Y|X)}_{\text{Fehlinformation}}$$

oder die Entropie der Quelle minus des Informationsverlusts

$$T(X;Y) = H(X) - \underbrace{H(X|Y)}_{\text{Informationsverlust}}$$

Fehlinformation größer als Informationsverlust (außer es passieren keine Fehler, oder die Entropie der Quelle ist maximal, dann $H(X|Y) = H(Y|X)$)

$\hookrightarrow$ Entropie am Ausgang des Kanals größer als die der Quelle (außer es passieren keine Fehler, oder die Entropie der Quelle ist maximal, dann $H(X) = H(Y) = 1$)

1.4. Kanalkapazität

Die Kanalkapazität ist das Maximum an Information, welche über den Kanal geht

$$C = \max(T(X;Y)) = \max(H(X) - H(X|Y))$$
$$= \max(H(Y) - H(Y|X))$$
$$= 1 - H(X|Y) = 1 - H(Y|X)$$

für binären, symmetrischen Kanal:

$$H(Y|X) = -p \cdot ld(p) - (1-p) \cdot ld(1-p)$$
$$C = 1 + p \cdot ld(p) + (1-p) \cdot ld(1-p)$$
$$C_S = C \cdot R_S$$

p: Bitfehlerwahrscheinlichkeit
C: Kanalkapazität in Information/Symbol $[C] = Bit/Symbol$
C_S: Kanalkapazität in Information/Zeit $[C_S] = Bit/s$
R_S: Symbolrate $[R_S] = Symbol/s$

Informationsfluss

$$H_S = H \cdot R_S$$

H_S: Informationsfluss in Information/Zeit $[H_S] = Bit/s$
H: Informationsgehalt pro Symbol $[H] = Bit/Symbol$
R_S: Symbolrate $[R_S] = Symbol/s$

1.5. Shannon-Theoreme

1.5.1. Erster Satz von Shannon

Quellcodierung

Ist ein Informationsfluss von H_S gewünscht, welcher kleiner als die Kanalkapazität C_S ist, ist es möglich einen Code zu finden, sodass der Informationsfluss übertragen werden kann

1.5.2. Zweiter Satz von Shannon

Kanalcodierung

Ist der die gewünschte Datenfluss R_S ($[R_S] = Symbol/s = bit/s$) kleiner als die Kanalkapazität C_S, dann gibt es einen Code, sodass die Fehlerwahrscheinlichkeit beliebig klein wird

1.5.3. Shannon-Hartley

Kanalkapazität abhängig vom SNR

$$C_S = W \cdot ld\left(1 + \frac{S}{N}\right)$$

C_S: Kanalkapazität in Information/Zeit $[C_S] = Bit/s$
W: Bandbreite $[W] = Hz$
S: Signalleistung
N: Rauschleistung

Bandbreiteneffizienz

Wird mit $R_S = C_S$ übertragen, kann die Bandbreiteneffizienz bestimmt werden

$$\frac{R_S}{W} = ld\left(1 + \frac{S}{N}\right)$$

$$S = \frac{E_b}{T_b} = E_b \cdot R_S \qquad N = N_0 \cdot W$$

$$\frac{S}{N} = \frac{E_b \cdot R_S}{N_0 \cdot W}$$

$\hookrightarrow$ damit S/N und Bandbreiteneffizienz abhängig von W

Ist eine Bandbreiteneffizienz gewünscht, kann das erfolderliche S/N ausgerechnet werden

$$\frac{E_b}{N_0} = \frac{W}{R_S}\left(2^{\frac{R_S}{W}} - 1\right)$$

$$\frac{E_b}{N_0}_{min} = -1,6\,dB$$

$\frac{R_S}{W}$: Bandbreiteneffizienz $[\frac{R_S}{W}] = \left(\frac{bit/s}{Hz}\right)$
E_b: Energie die für die Übertragung von 1 bit aufgewendet wird $[E_b] = J = Ws = W/Hz$
T_b: Zeitdauer eines bits $[T_b] = s = 1/Hz$

2. Blockcodes

Code beschrieben durch $C(n, k, d)$

n: Länge Codewort
k: Länge Informationswort
d: Mindestabstand

Coderate: $CR = \frac{k}{n}$

2.1. Generelles

Generatormatrix

Erzeugung eines Codewortes über Multiplikation eines Informationsvektors

$\vec{c} = i \cdot G$

c: Codewort
i: Informationswort
G: Generatormatrix

Prüfmatrix

$H\vec{c}^T = 0$

$H \cdot G^T$

H: Prüfmatrix

Syndrom

wenn Empfangswort r fehlerhaft ist (und nicht zum Coderaum gehört) dann ist das Produkt aus Prüfmatrix und Empfangswort das *Syndrom* und nicht mehr 0

$H\vec{r}^T = H(\vec{c} + \vec{f})^T = H\vec{f}^T = \vec{s} \neq 0$

Systematische Codes

Systematischer Code: Einheitsmatrix I ist Teil der Generatormatrix

$G = [I|G']$

Bsp.: C(7, 4, 3): $G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$

Lineare Codes

Eigenschaften:

- Mindestgewicht = Mindestabstand
- Codewort + anderes Codewort = wieder Codewort
- Nullwort ist teil des Codes

Gewicht

Gewicht eines Codewortes: Anzahl der von 0 verschiedenen Stellen

Mindestgewicht: Minimale Anzahl an Stellen, die von 0 verschieden sind

Distanz/Abstand

Hamming-Distanz: Anzahl verschiedener Stellen zweier Codewörter

Mindestdistanz: Mindestanzahl verschiedener Stellen zweier beliebiger Codewörter eines Codes

Fehlerkorrektur/Fehlererkennung

Anzahl der korrigierbaren Fehler e bei gegebenem Abstand d

$e = \left\lfloor \frac{d-1}{2} \right\rfloor$

bei ungeradem d:

es werden $d - 1$ Fehler erkannt

bei geradem d:

es werden alle ungeraden Anzahlen an Fehlern erkannt

2.2. Fehlerwahrscheinlichkeit

Restblockfehlerwahrscheinlichkeit

Wahrscheinlichkeit, dass von n Symbolen beliebige m falsch und die übrigen Symbole $n - m$ richtig sind:

$$\binom{n}{m} \cdot p^m \cdot (1-p)^{n-m}$$

für ungerades d:

alle Fehler werden entweder korrigiert, oder nicht erkannt

Kann ein Code e Fehler korrigieren, dann verbleibt eine Restblockfehlerwahrscheinlichkeit von:

$$P_{Block} = \sum_{m=e+1}^{n} \binom{n}{m} \cdot p^m \cdot (1-p)^{n-m}$$

$$= 1 - \sum_{m=0}^{e} \binom{n}{m} \cdot p^m \cdot (1-p)^{n-m}$$

p: Fehlerwahrscheinlichkeit vor der Decodierung
P_{Block}: Wahrscheinlichkeit, für Symbolfehler im decodierten Block

für gerades d:

alle ungeraden Fehler werden erkannt, da sie nicht in Korrekturkugeln liegen; können damit aber auch nicht korrigiert werden

$\rightarrow$ für die Wahrscheinlichkeit, dass Fehler nicht erkannt werden, tragen also nur die geraden Fehleranzahlen, welche größer als e sind, bei

$$P_{Block,erkennen} = \sum_{m=e+1}^{n} \binom{n}{2m} \cdot p^{2m} \cdot (1-p)^{n-2m}$$

P_{Block} gleich wie bei gerader Anzahl

Symbolfehlerwahrscheinlichkeit

Block besteht aus k Symbolen

Bei gegebener Restblockfehlerwahrscheinlichkeit, gibt es eine Symbolfehlerwahrscheinlichkeit von

$$P_{Symb} = P_{S|Block} \cdot P_{Block} = \frac{2^{k-1}}{2^k - 1} P_{Block}$$

2.3. Hamming-Codes

immer $d = 3$, damit immer $e = 1$

$n = 2^h - 1$

$k = n - h$

h	n	k	d
2	3	1	3
3	7	4	3
4	15	11	3
5	31	26	3
⋮	⋮	⋮	⋮

Konstruktion

1. Erstellung Prüfmatrix

Spalten sind Dualdarstellung der Spaltennummer

H_{n-kxn}

Beispiel: C(7, 4, 3)

$$H_{3x7} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

2. Spalten tauschen, sodass hinten Einheitsmatrix

$H = [A_{n-kxk} | I_{n-kxn-k}]$

Spalte 1 mit 7
2 mit 6
4 mit 5

$$\rightarrow H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} = [A_{3x4} | I_{3x3}]$$

3. Generatormatrix aufstellen

$G_{kxn} = [I_{kxk} | -A^T_{kxn-k}]$

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

systematisch, da Informationswort zusammenhängend im Codewort steht

4. Rücktauschen der Spalten

Spalte 1 mit 7
2 mit 6
4 mit 5

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

quasi-systematisch, da Informationswort zwar im Codewort, aber nicht zusammenhängend

Fehler und Syndrom

Fehler gibt an, an welcher Stelle des Empfangsworts ein Fehler aufgetreten ist

Bsp.:

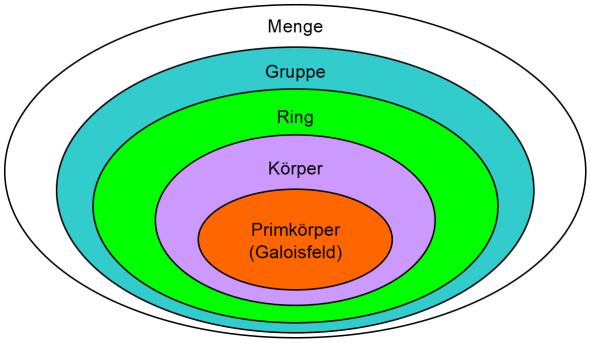
$$\vec{s} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

$\rightarrow$ Fehler an Stelle $(011)_b = 3$ im Empfangswort

$\rightarrow$ da binärer Code, muss für Fehlerkorrektur das dritte bit nur invertiert werden

3. Galois-Felder

3.1. Algebraische Strukturen



Menge

Verbund von Elementen, welche keine Operationen beinhalten (Möbel können eine Menge sein, es kann aber nicht Tisch + Stuhl gerechnet werden)

Halbgruppe

Menge A mit Verknüpfung »+« ist eine Halbgruppe, wenn

- Abgeschlossenheit (+ zweier Elemente von A ergibt wieder ein Element von A)
- Assoziativität (Reihenfolge der Operation mit + spielt keine Rolle, $a + (b + c) = (a + b) + c$)
- Existenz eines neutralen Elements (Element a + neutrales Element n ergibt wieder Element a)

Gruppe

Halbgruppe plus

- Existenz eines additiven inversen Elements ($a + b = n$)

Abelsche oder kommutative Gruppe

Gruppe plus

- Kommutativität (Reihenfolge der Operanden spielt keine Rolle, $a + b = b + a$)

Ring

abelsche Gruppe plus

- Abgeschlossenheit bezüglich »·«
- Assoziativität bezüglich »·«
- Distributivität ($a \cdot (b + c) = a \cdot b + a \cdot c$)

Körper

Ring plus

- Kommutativität bezüglich »·«($a \cdot b = b \cdot a$)
- Neutrales Element bezüglich »·«
- Inverses Element bezüglich »·«für jedes Element

Primkörper/Galois-Feld

Körper, indem Addition und Multiplikation mod p gerechnet wird (p muss dabei eine Primzahl sein)
$\hookrightarrow GF(p)$

3.2. Eigenschaften Galois-Felder

Primitives Element

Element α, welches durch ihre $p-1$ Potenzen alle Elemente (außer 0) des $GF(p)$ erzeugt

Bsp. $GF(5), \alpha = 2$:

$2^0 = 1 \mod 5 = 1$
$2^1 = 2 \mod 5 = 2$
$2^2 = 4 \mod 5 = 4$
$2^3 = 8 \mod 5 = 3$

ab hier zykische Wiederholung:
$2^4 = 16 \mod 5 = 1$

Polynome

Folge an n Zahlen im Galois-Feld wird als Polynom vom Grad $n-1$ geschrieben

$\hookrightarrow \{1;4;3;1\} \rightarrow A(x) = 1x^3 + 4x^2 + 3x + 1$

Auswertung des Polynoms $A(x)$ an verschiedenen Stellen von α^i ergibt ihre Fouriertransformierte $a(x)$

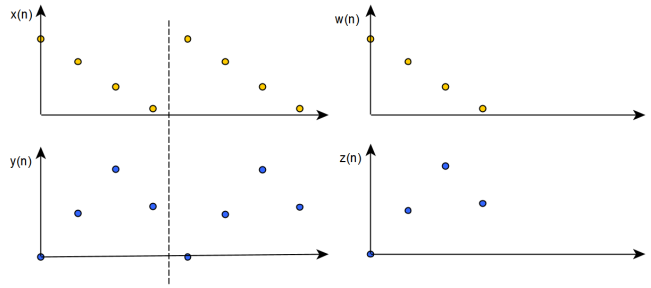
$a_i = A(\alpha^i)$

Zyklische Faltung

Polynommultiplikation im Galois-Feld $\rightarrow$ zyklische Faltung

Normale Faltung mit endlichen Signalen $\rightarrow$ endliches Faltungsergebnis

Zyklische Faltung: Signale sind periodisch, damit Faltungsergebnis ebenfalls periodisch (und damit unendlich lang)



links: zyklische Faltung rechts: normale Faltung

4. Reed-Solomon-Code

4.1. Wunsch und Idee

Wunsch

Konstruktion eines Codes mit vorgegebener Korrekturfähigkeit $\rightarrow$ Vorgabe des Mindestabstandes d

$e = \left\lfloor \frac{d-1}{2} \right\rfloor$
$d = 2e + 1$

bei linearem Code ist Mindestabstand = Mindestgewicht

$\rightarrow$ Codeworte haben mind. d von 0 verschiedene Koeffizienten

d'Alembert: Polynom vom Grad n hat n komplexe (oder höchstens n reelle) Nullstellen; auch im Galois-Feld

Idee

Konstruktion des Informationswortes als Polynom $A(x)$ mit Grad $k-1$ (damit höchstens $k-1$ Nullstellen)

Im $GF(p)$ mit Ordnung $n = p-1$ kann man $A(x)$ an n Stellen auswerten, danach wiederholen sich die Werte

$\rightarrow$ Auswertung des Polynoms für verschiedene x (bzw. α^i) ergeben die Koeffizienten a_i des Polynoms $a(x)$

$a_i = A(\alpha^i)$ **IDFT**

von diesen sind höchstens $k-1$ Null (weil $grad(A(x)) = k-1$) von diesen sind also mind. $n-(k-1)$ von Null verschieden $\rightarrow$ Mindestgewicht d

$d = n - (k-1) = n - k + 1$

Rücktransformation des Codeworts in Informationswort:

$A_i = n^{-1} a(\alpha^{-i})$ **DFT**

4.2. Codierung

Verschiedene Möglichkeiten aus einem Informationswort ein Codewort zu generieren

4.2.1. Generatorpolynom

Erzeugt zusammenhängende Nullstellen im Codewort = Syndromstellen

$$g(x) = \prod_{i=k}^{n-1} \left(x - \alpha^{-i}\right)$$

Syndromstellen beginnen hier bei k, es sind aber alle anderen Stellen möglich, solange sie zusammenhängen

$grad(g(x)) = d - 1 = n - k =$ Anzahl Syndromstellen

$g(x)$: Generatorpolynom
$i(x)$: Informationspolynom

4.2.2. Prüfpolynom

Prüfpolynom:

$$h(x) = \prod_{i=0}^{k-1} \left(x - \alpha^{-i}\right)$$

Produkt aus Generator- und Prüfpolynom ist 0

$g(x) \cdot h(x) = 0$

und Produkt aus Codepolynom und Prüfpolynom ist 0

$a(x) \cdot h(x) = 0$

genau da, wo $g(x)$ (oder $a(x)$) Nullstellen hat (also G_i 0 ist) hat das Prüfpolynom $h(x)$ keine Nullstellen (ist also H_i nicht 0) und umgekehrt

4.2.3. IDFT (nicht systematisch)

$a_i = A(\alpha^i)$

$A(x)$: Informationswort
a_i: Koeff. des Codewortes

4.2.4. Polynommultiplikation (nicht systematisch)

$a_i = g(x) \cdot i(x)$

4.2.5. Polynomdivision (systematisch)

Informationswort ist Teil des Codewortes (an den hohen Potenzen)

$a^*(x) = i_{k-1}x^{n-1} + i_{k-2}x^{n-2} + ... + i_1x^{n-k+1} + i_0x^{n-k}$

jedes Codewort muss durch Generatorpolynom teilbar sein → ist für $a^*(x)$ i.A. nicht der Fall

$$\frac{a^*(x)}{g(x)} = b(x) + \frac{rest(a^*(x))}{g(x)}$$
$$\rightarrow \frac{a^*(x) - rest\,(a^*(x))}{g(x)} = b(x)$$
$$a(x) = a^*(x) - rest\,(a^*(x))$$

$rest\,(a^*(x))$: Divisionsrest

4.2.6. Zyklischer Code

Multiplikation eines Polynoms mit x^i verschiebt Koeff. des Polynoms um i-Stellen

durch mod-Rechnung des Exponenten verschieben sich höhere Exponenten wieder an den Anfang des Polynoms

Bsp.:

$x \cdot a(x) = x \cdot (2x^2 + x + 1) = 2x^3 + x^2 + x = x^2 + x + 2$

4.3. Decodierung

Idee:

Addition des Fehlerpolynoms $f(x)$ mit t Koeffizienten (d.h. t Fehler sind auf dem Kanal aufgetreten) zum gesendeten Codewort $a(x)$

im Zeitbereich:

$r(x) = a(x) + f(x)$

im Frequenzbereich:

$R(x) = A(x) + F(x)$

gedanklich wird ein Polynom $c(x)$ aufgestellt, welches t Nullen an den Fehlerstellen hat

Da die Koeffizienten von $c(x)$ die Auswertung ihrer Fouriertransformierten $C(x)$ ist, ist der Grad von $C(x)$ t

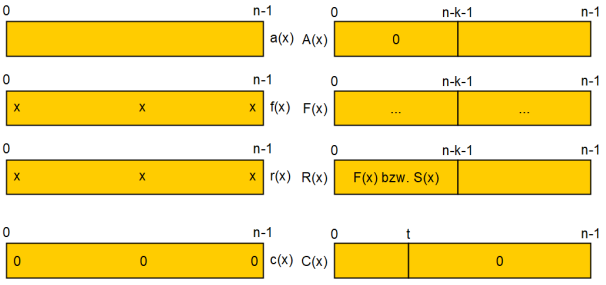
Da $c(x)$ gerade dort 0 ist, wo $f(x)$ ungleich 0, ist das Produkt $f_i \cdot c_i$ immer 0 (Achtung, keine Polynommultiplikation gemeint, sondern punktweise Multiplikation)

$f_i \cdot c_i = 0$

wenn Zeitbereich = 0 → Frequenzbereich = 0

$F(x) \cdot C(x) = 0$

Achtung: hier Polynommultiplikation/ Faltung/ Filterung gemeint
↪ Aufstellen der Schlüsselgleichungen



4.3.1. Vorgehen

1. Fouriertransformation des empfangenen Codewortes $r(x) \rightarrow R(x)$
2. Auslesen der Koeff. des Syndrompolynoms $(S_0, ..., S_n)$ aus $R(x)$ und Aufstellen des Syndrompolynoms
3. Berechnung des $C(x)$ aus Schlüsselgleichungen oder euklidschem Algorithmus
4. Berechnung der Fehlerstellen durch Nullstellensuche von $C(x)$
5. Berechnung des Fehlerwertes über Schlüsselgleichungen oder Forney-Algorithmus

4.3.2. Schlüsselgleichungen

beschreiben, dass Faltung von $C(x)$ und $F(x)$ Null ist (Achtung: zyklische Faltung, siehe Abschnitt 3)

F_0 bis F_{n-k-1} (bzw. F_{d-2}) sind bekannt, da diese direkt an den Syndromstellen von $R(x)$ stehen

Alle C-Koeff. sind unbekannt, außer C_t, dieser wird zu 1 gesetzt

$C_t = 1$

da Anzahl der Fehler (t) unbekannt ist, muss ausprobiert werden, welche minimale Anzahl an Fehlern die Schlüsselgleichungen widerspruchsfrei erfüllt

Lösen der Schlüsselgleichungen nach $C(x)$

↪ Nullstellensuche von $C(x)$ ergibt die Nullen des $c(x)$

↪ wenn Grad von $C(x)$ nicht mit Anzahl der Nullstellen übereinstimmt → Decodierversagen

Lösen der Schlüsselgleichungen nach $F(x)$

↪ $f(x)$ aus Rücktransformation von $F(x)$

↪ $f(x)$ von $r(x)$ abziehen, man erhält $a(x)$

$a(x) = r(x) - f(x)$

4.3.3. Euklidscher Algorithmus

Suche des ggT zweier Zahlen

Kann zur Lösung der Schlüsselgleichungen verwendet werden

Rest:

$r_n = v_n a_n + w_n b_n$

Rekursionsformeln für v_n und w_n:

$v_n = v_{n-2} - q_n v_{n-1}$

$w_n = w_{n-2} - q_n w_{n-1}$

q_n: Quotient des vorherigen Schrittes

Initialisierung:

$v_{-1} = 1 \quad v_0 = 0$
$w_{-1} = 0 \quad w_0 = 1$

Suche des $C(x)$ und damit den Fehlerstellen

Polynomdivision von x^{d-1} und des Syndrompolynoms $S(x)$

$x^{d-1} : S(x)$

Wenn Rest der Division im Grad nicht kleiner ist als die Anzahl der Fehler e, die maximal korrigiert werden können → weiter: $S(x) : r_1(x)$

usw.

ist Grad des Restes kleiner als e → Berechnung des $C(x)$ und des $T(x)$

$\hookrightarrow C(x) = w_n$

$\hookrightarrow T(x) = -r_n$

4.3.4. Forney-Algorithmus

Fehlerstellenberechnung durch Nullstellensuche in $C(x)$

$c_i = C(\alpha^i)$

Fehlerwertberechnung aus gegebenem $C(x)$ und $T(x)$

$$f_i = x^q \cdot n \cdot x^{-1} \left. \frac{T(x)}{C'(x)} \right|_{x=\alpha^i}$$

q: Verschiebung der Syndromstellen ($q = 5$, wenn Syndrom an Stelle 5)

1. Hinweis: Fehlerwert an den Stellen, an dem keine Fehler passiert sind, ist im Allgemeinen nicht 0

2. Hinweis: Ableitungsregeln beachten, Exponent der beim Ableiten als Faktor vorgezogen wird, ist Teil des Grundkörpers und nicht des Erweiterungskörpers (siehe Abschnitt A.2)

3. Hinweis: n ist auch Teil des Grundkörpers und wird deshalb auch mod 2 gerechnet

5. Erweiterungskörper

5.1. Idee

Erweitern des Grundkörpers (z.B. 2) mit Exponent (z.B. 4) → $GF(2^4)$

Irreduzibles Polynom ist die Primzahl des Erweiterungskörpers z.B. in $GF(2^4)$:
$p(x) = x^4 + x + 1$

Irreduzibles Polynom: $ggT(p(x), b(x)) = 1$

größter gemeinsamer Teiler mit einem beliebigen Polynom $b(x)$ ist 1

d.h. $p(x)$ kann nicht in Linearfaktoren zerlegt werden

für irreduzible Polynome gilt:
- ist durch kein Polynom ohne Rest teilbar
- hat keine Nullstellen

aber: Nullstellen sind wichtig für Nutzung des RS-Codes, daher »Erfindung« des Elements α, welches Nullstelle von $p(x)$ ist

$p(\alpha) = 0$

am Beispiel:
$p(\alpha) = \alpha^4 + \alpha + 1 = 0$

Analogie: »Erfindung« von j, sodass gilt:

$j^2 + 1 = 0$

primitives Polynom: Nullstelle (α) des primitiven Polynoms erzeugt alle Elemente (außer 0) des Erweiterungskörpers

primitives Element: Nullstelle α des primitiven Polynoms

5.2. Eigenschaften von Erweiterungskörpern

Ordnung des primitiven Elements: $2^m - 1$ im $GF(2^m)$

Erzeugung der Elemente über Potenzieren des primitiven Elements α

zum Körper $GF(2^m)$ gehören 2^m Elemente ($2^m - 1$ dieser wird durch Potenzieren von α erzeugt)

Elemente der Erweiterungskörper sind Polynome

Darstellung

Erzeugung von bspw. α^3 in $GF(2^4)$ mit irreduziblem Polynom $p(x) = x^4 + x + 1$:

$\alpha^3 = 1 \cdot \alpha^3 + 0 \cdot \alpha^2 + 0 \cdot \alpha^1 + 0 \cdot \alpha^0$

dazugehörige Binärdarstellung:

1000

5.3. Kürzere Codes

Verkürzung

Streichen von Informationswortstellen und Codewortstellen

Distanz und damit Fehlerkorrigierbarkeit bleibt gleich

Code ist nicht mehr zyklisch

Bsp.: Verkürzung eines $C(6, 2, 5)$ um 1 auf $C(5, 1, 5)$

Äquivalent zu Einfügen von Nullen an den hohen Potenzen des Informationswortes

Punktierung

Streichen bestimmter Stellen aus dem Codewortbitstrom

Mindestabstand bleibt im Allgemeinen nicht erhalten

6. BCH-Codes

6.1. Idee

Wunsch: reelle Koeffizienten (reell im $GF(2^m)$ = binär)

Für Erweiterungskörper war α die Nullstelle des primitiven Polynoms

ABER: d'Alembert: Polynom vom Grad m hat m Nullstellen

Wo sind die restlichen Nullstellen der primitiven Polynome höheren Grades?

$\hookrightarrow$ wenn α Nullstelle von $p(x)$ ist, dann sind auch $\alpha^2, \alpha^{2^2}, \alpha^{2^3}, ..., \alpha^{2^{m-1}}$ Nullstellen ($\rightarrow$ konjugiert komplexe Nullstellen)

Ein Polynom aus konjugiert komplexen Nullstellen hat reelle (in unserem Fall binäre) Koeffizienten

im Allgemeinen sind $\alpha^{j \cdot 2^i \mod 2^m - 1}$ $(i = 0, ..., m-1)$ konjugiert komplexe Nullstellen

$\hookrightarrow$ Kreisteilungsklassen

6.2. Kreisteilungsklassen

Sind nur vom Erweiterungskörper abhängig, nicht vom primitiven Polynom

Eine Kreisteilungsklasse enthält die Exponenten der konjugiert komplexen Nullstellen des primitiven Polynoms

$K_j = j \cdot 2^i \mod 2^m - 1 \qquad (i = 0, ..., m-1)$

Bsp.: $GF(2^3)$

$K_0 = \{0\}$
$K_1 = \{1, 2, 4\}$
$K_2 = \{2, 4, 1\} = K_1$
$K_3 = \{3, 6, 5\}$

D.h. um ein (Generator)polynom aufzustellen, welches reelle (=binäre) Koeffizienten hat, braucht man alle konjugiert komplexen Nullstellen

6.2.1. Generatorpolynom

Konstruktion eines Codes über Vorgabe der Korrekturfähigkeit e und damit d

RS-Codes: Generatorpolynom vom Grad $d-1$ (siehe Abschnitt 4.2.1)

$\hookrightarrow$ d.h. $d-1$ zusammenhängende Nullstellen

Zusammenhängende Nullstellen bei BCH-Codes nur möglich, wenn alle Elemente der entsprechenden Kreisteilungsklassen im Generatorpolynom vorhanden sind

7. Faltungscodes

Filterung der Eingangssequenz mit FIR-Filter

Beschreibung durch $C(n, k, [z])$

n: Anzahl Ausgänge
k: Anzahl Eingänge
z: Anzahl an Speicherzellen

Generatormatrix gibt Verhalten des Coders vollständig an

$G = \begin{pmatrix} 142 \\ 345 \\ 711 \end{pmatrix}_O$ Zahlen in Oktaldarstellung

$\rightarrow G = \begin{pmatrix} 001\,100\,010 \\ 011\,100\,101 \\ 111\,001\,001 \end{pmatrix}$

Freie Distanz

Gewicht (Anzahl an Einsen) der Ausgangssequenz um vom Zustand »0« wieder zurück in den Zustand »0« zu kommen

7.1. Codierung

7.1.1. Ein Ausgang

Faltungscoder ohne Redundanz

Normaler FIR-Filter mit binären Koeffizienten, Generatorsequenz ist Impulsantort

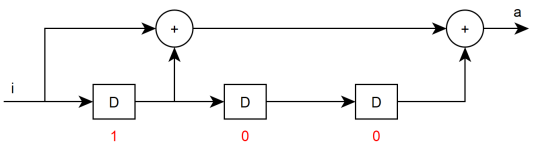
Delay: $z^{-1} = D$

Impulsantort: $\vec{g}$

Inhalt der Speicherzellen: $\vec{d}$

Ausgang: $a = \vec{g}^T \cdot \begin{pmatrix} i \\ \vec{d} \end{pmatrix}$

Beispiel



Impulsantort: $g = 1 + 1D + 0D^2 + D^3 = 1 + D + D^3$

Generatorsequenz: $\vec{g}^T = (1 \quad 1 \quad 0 \quad 1)$

Von links nach rechts steht (100) in den Speicherzellen und es wird $i = 1$ hineingeschrieben

$\hookrightarrow \vec{d} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

$$a = (1 \quad 1 \quad 0 \quad 1) \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = 0$$

7.1.2. Mehrere Ausgänge

Filter mit mehreren Ausgängen

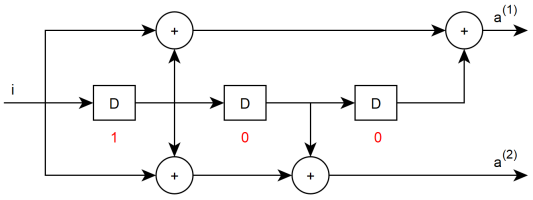
n Impulsantorten des Filters geben n verschiedene Ausgänge

Generatorsequenz wird zu Generatormatrix mit n Generatorsequenzen

$$\vec{a} = G^T \cdot \begin{pmatrix} i \\ \vec{d} \end{pmatrix} = \begin{pmatrix} a^{(1)} \\ a^{(2)} \end{pmatrix} \qquad G^T = \begin{pmatrix} \vec{g_1}^T \\ \vec{g_2}^T \end{pmatrix}$$

Ausgangssequenz: $a = a_1^{(1)}, a_1^{(2)}, a_2^{(1)}, a_2^{(2)}, ...$

Beispiel



Im Speicher steht wieder von links nach rechts (100) und es wird $i = 1$ hineingeschrieben

$$\vec{a} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

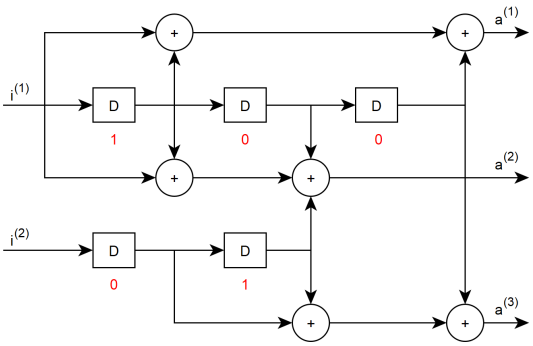
7.1.3. Mehrere Eingänge

Mehrere Eingänge $(i^{(1)}, i^{(2)})$ um Coderate anzupassen

für jeden Ausgang gibt es jeweils n Generatorsequenzen (g_{ij})

i: Eingang
j: Ausgang

Beispiel



$\vec{g_{11}}^T = (1 \quad 1 \quad 0 \quad 1)$ $\qquad$ $\vec{g_{21}}^T = (0 \quad 0 \quad 0 \quad 0)$

$\vec{g_{12}}^T = (1 \quad 1 \quad 1 \quad 1)$ $\qquad$ $\vec{g_{22}}^T = (0 \quad 0 \quad 1 \quad 0)$

$\vec{g_{13}}^T = (0 \quad 0 \quad 0 \quad 1)$ $\qquad$ $\vec{g_{23}}^T = (0 \quad 1 \quad 1 \quad 0)$

Zusamenfassung in G

$$G = \overbrace{\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}}^{\begin{matrix} E1 & E2 & E1 & E2 & E1 & E2 & E1 & E2 \end{matrix}} \begin{matrix} \} \text{Ausgang 1} \\ \} \text{Ausgang 2} \\ \} \text{Ausgang 3} \end{matrix}$$
$$\underbrace{\qquad\qquad}_{D^0} \quad \underbrace{\qquad\qquad}_{D^1} \quad \underbrace{\qquad\qquad}_{D^2} \quad \underbrace{\qquad\qquad}_{D^3}$$

7.2. Modifikationen

Punktierung

Streichen bestimmmter Stellen aus dem Codewort

Punktierung über Punktierungsmatrix gegeben

Bsp.: für einen Coder mit 2 Ausgängen

$$G = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{matrix} \} \text{Ausgang 1} \\ \} \text{Ausgang 2} \end{matrix}$$

Ausgangssequenz a punktieren durch bitweise AND mit der Matrixssequenz $p = (1\ 1\ 0\ 1)$:

$a = \quad 1001\ 0100\ 1011\ 1001\ 0100$
$p = \quad 1101\ 1101\ 1101\ 1101\ 1101$

$a^* = 10x1\ 01x0\ 10x1\ 10x1; 01x0$

x: gestrichene Stelle, wird nicht übertragen (wird im Trellis immer als Fehler gewertet)

Terminierung

Endzustand des Coders bekannt, durch Anhängen von z Nullen

$\hookrightarrow$ damit ist garantiert, dass im Coder am Ende nur Nullen stehen

Tail-Biting

Endzustand nach Codierung eines Informationswortes wird als Anfangszustand für eine erneute Codierung verwendet (bei FIR-Filtern ist eine erste Codierung nicht notwendig, da der Endzustand nur von den letzten z Informationsbits abhängt)

$\hookrightarrow$ damit ist der Anfangszustand identisch mit Endzustand

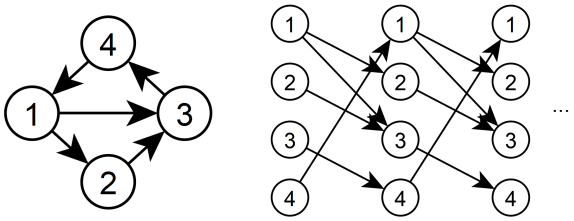
$\hookrightarrow$ der Decoder decodiert und kennt mit großer Sicherheit den Anfangszustand und damit auch den Endzustand

7.3. Decodierung

Mit Viterbi-Algorithmus

Konstruktion eines Zustandsgraphen für den Coder

Aus Zustandgraph wird Trellis-Diagramm erstellt



gesucht: Informationsfolge, welche in den Coder geschickt wurde

Metrik: Anzahl der übereinstimmenden Ausgangsbits

Pfad mit höchster Übereinstimmung wird genommen

Außer:

- bei Terminierung: Endzustand 0
- bei Tail-Biting: Endzustand = Anfangszustand

A. Hilfreiches

A.1. Inverse in Galois-Feldern

Additive Inverse

gegeben: -3 in $GF(5)$

gesucht: additive Inverse

bedeutet: $3 + x \mod 5 = n = 0$

n: neutrales Element der Addition ($= 0$)

$x = 2$, da $3 + 2 = 5$ und $5 \mod 5 = 0$

daher: $-3 = 2$

oder: mit Tabelle

gegebene Zahl als Index behandeln, passenden Wert raussuchen

Index	-3	-2	-1	0	1	2	3	4	5	6
Wert	2	3	4	0	1	2	3	4	0	1

Multiplikative/modulare Inverse

gegeben: 2^{-1} in $GF(7)$

gesucht: multiplikative Inverse

bedeutet: $2^1 \cdot x \mod 7 = n = 1$

n: neutrales Element der Multiplikation ($= 1$)

$x = 4$, da $2^1 \cdot 4 = 8$ und $8 \mod 7 = 1$

daher: $2^{-1} = 4$

oder: mit Logarithmentafel

gegebene Potenz als Index behandeln, passenden Wert raussuchen

Index	-3	-2	-1	0	1	2	3	4	5	6
Wert	1	2	4	1	2	4	1	2	4	1

A.2. Rechnen im Erweiterungskörper

alle Rechnungen am Beispiel: $GF(2^4)$ mit $p(x) = x^4 + x + 1$

Addition

entweder mit Additionstabelle, oder binär stellenweise XOR rechnen (ohne Übertrag)

$\alpha^8 + \alpha^{13} = 5 + 13 = 0101 \oplus 1101 = 1000 = 8 = \alpha^3$

Multiplikation

entweder mit Multiplikationstabelle, oder addieren der Exponenten

$\alpha^6 \cdot \alpha^9 = \alpha^{15} = \alpha^0 = 1$

$\frac{4}{12} = \frac{\alpha^2}{\alpha^6} = \alpha^2 \cdot \alpha^{-6} = \alpha^2 \cdot \alpha^{-6} \cdot \alpha^{15} = \alpha^{11}$

Ableiten

Wichtig: Die Exponenten, welche beim Ableiten als Faktor runtergezogen werden, sind keine Elemente des Erweiterungskörpers, sondern des Grundkörpers $\rightarrow \mod 2$ rechnen!

$f(x) = x^3 + 3x^2 + 4 = \alpha^0 x + \alpha^4 x^2 + \alpha^2$

$\hookrightarrow f'(x) = 3 \cdot x^2 + 2 \cdot \alpha^4 x = x^2$

A.3. Syndromstellen aus Generatorpolynom

Bsp.: $GF(5)$ mit $\alpha = 2$

gegeben $g(x) = x^2 + 5x + 4$

$\hookrightarrow$ 2 Syndromstellen

Suchen über stures Einsetzen der Elemente des $GF(5)$:

$g(\alpha^{-0}) = \alpha^{-0^2} + 3\alpha^{-0} + 2 = 1 + 3 + 2 = 1$

$\hookrightarrow$ Position 0 keine Syndromstelle

$g(\alpha^{-1}) = \alpha^{-1^2} + 3\alpha^{-1} + 2 = 3^2 + 3 \cdot 3 + 2 = 0$

$\hookrightarrow$ Position 1 ist Syndromstelle

$g(\alpha^{-2}) = \alpha^{-2^2} + 3 \cdot \alpha^{-2} + 2 = 4^2 + 3 \cdot 3^2 + 2 = 0$

$\hookrightarrow$ Position 2 ist Syndromstelle

A.4. ABC/PQ-Formel

ABC: $ax^2 + bx + c = 0$

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

PQ: $x^2 + px + q$

$$x_{1,2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$$

B. Polynome

B.1. Polynommultiplikation

B.2. Polynomdivision

allgemein:

gegeben: $(6x^3 - 2x^2 + x + 3) : (x^2 - x + 1)$

$$\begin{array}{rl} 6x^3 - 2x^2 + x + 3 & \big| \; x^2 - x + 1 \\ \underline{-6x^3 + 6x^2 - 6x} & \big| \; 6x + 4 \\ 4x^2 - 5x + 3 & \\ \underline{-4x^2 + 4x - 4} & \\ -x - 1 & \end{array}$$

Quotient $q(x) = 6x + 4$

Rest $r(x) = -x - 1$

Horner-Schema

zur Polynomdivision mit Linearfaktor

Rest der Division mit $(x - x_0)$ ist Wert des Polynoms an der Stelle x_0

gegeben: $p(x) = 3x^3 + 2x^2 - 5x - 10$

	x^3	x^2	x^1	x^0	2
	3	2	-5	-10	
		6	16	22	
	3	8	11	12	

Quotient: $q(x) = 3x^2 + 8x + 11$

Rest: $r(x) = 12 = p(2)$

C. Lineare Algebra

Matrix-Multiplikation

$$A \cdot B = \begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix}_{3,2} \begin{pmatrix} g & h \\ i & j \end{pmatrix}_{2,2} = \begin{pmatrix} ag + bi & ah + bj \\ cg + di & ch + dj \\ eg + fi & eh + fj \end{pmatrix}_{3,2}$$

$$\vec{v} \cdot \vec{w}^T = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \cdot \begin{pmatrix} d & e & f \end{pmatrix} = \begin{pmatrix} a \cdot d & a \cdot e & a \cdot f \\ b \cdot d & b \cdot e & b \cdot f \\ c \cdot d & c \cdot e & c \cdot f \end{pmatrix}$$

Skalarprodukt:

$$\vec{v}^T \cdot \vec{w} = \begin{pmatrix} a & b & c \end{pmatrix} \cdot \begin{pmatrix} d \\ e \\ f \end{pmatrix} = a \cdot d + b \cdot e + c \cdot f$$

bei komplexen Vektoren: $\vec{v}^H \cdot \vec{w}$

Transponieren

Zeilen werden Spalten, Spalten werden Zeilen

$$A^T = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}^T = \begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix}$$

Invertieren

für 2x2-Matrizen:

$$A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -c \\ -b & a \end{pmatrix}$$

Diagonale Matrix

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

Spalten der Matrix sind Eigenvektoren

1; -5; 3 sind die Eigenwerte der Eigenvektoren

Hermitesche Matrix

nur für quadratische Matrizen

$$A = A^H = (A^*)^T = \begin{pmatrix} 1 & 5 - j & 3j \\ 5 + j & 2 & 3 - 2j \\ -3j & 3 + 2j & 3 + 4j \end{pmatrix}$$

Eigenvektoren von hermiteschen Matrizen sind orthogonal

Unitäre Matrix

$A \cdot A^H = k \cdot I$

k: Skalierungsfaktor (bei skaliert unitären Matrizen)

Toeplitz-Struktur

Eine Matrix hat Toeplitz-Struktur, wenn alle Diagonalen parallel zur Hauptdiagonalen, die gleichen Elemente enthalten:

$$T = \begin{pmatrix} 0 & -2 & -5 & -3 \\ 1 & 0 & -2 & -5 \\ 2 & 1 & 0 & -2 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

Hermitesche Toeplitz-Matrizen sind positiv oder negativ definit, abhängig vom Vorzeichen der Elemente auf der Hauptdiagonalen

Vandermonde-Matrix

Spalten: Indezes gleich
Zeilen: Potenzen gleich

$$S = \begin{pmatrix} 1 & 1 & ... & 1 \\ x_0^1 & x_1^1 & ... & x_{N-1}^1 \\ x_0^2 & x_1^2 & ... & x_{N-1}^2 \\ ... & ... & ... & ... \\ x_0^{N-1} & x_1^{N-1} & ... & x_{N-1}^{N-1} \end{pmatrix}$$

Determinante

nur für quadratische Matrizen

für 2 × 2-Matrix:

$$det(A) = det\left[\begin{pmatrix} a & b \\ c & d \end{pmatrix}\right] = a \cdot d - b \cdot c$$

für 3 × 3-Matrix:

$$det(C) = det\left[\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}\right]$$

$$= a \cdot det\left[\begin{pmatrix} e & f \\ h & i \end{pmatrix}\right] - b \cdot det\left[\begin{pmatrix} d & f \\ g & i \end{pmatrix}\right] + c \cdot det\left[\begin{pmatrix} d & e \\ g & h \end{pmatrix}\right]$$

Rang einer Matrix

Eine Matrix hat vollen Rang, wenn die Determinante ungleich 0 ist

Ist die Determinante gleich 0, ist die Matrix/das Gleichungssystem überbestimmt

Eigenvektoren/ Eigenwerte

Eigenvektoren einer Matrix werden bei einer Matrixtransformation nur in ihrer Länge geändert, nicht in ihrer Richtung

Faktor, um den ein Eigenvektor gedehnt oder gestaucht wird, ist der zum Eigenvektor zugehöriger Eigenwert λ

$$A\vec{v} = \lambda\vec{v}$$

$$(A - \lambda I)\vec{v} = \vec{0}$$

A: Matrix
$\vec{v}$: Eigenvektor
λ: Eigenwert(e)
I: Einheitsmatrix

Eigenwerte von positiv (oder negativ) definiten Matrix sind immer positiv (oder negativ)

D. Digitale Signalverarbeitung

Diskretisierung und Fensterung

Diskretisierung ○—● Periodische Fortsetzung
Diskretisierung ●—○ Periodische Fortsetzung
Begrenzung → Leck-Effekt

Fourier-Transformation (kontinuierlich)

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft}\, dt$$

DFT

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi \frac{nk}{N}}$$

n: Frequenzindex
k: Zeitindex

Auflösung DFT

$$\Delta f = \frac{f_a}{N} = \frac{1}{t_a \cdot N} = \frac{1}{\Delta t}$$

Δf: spektrale Auflösung
f_a: Abtastfrequenz
t_a: Abtastrate
N: Anzahl Abtastwerte
Δt: Messdauer

Fensterung

Fensterung im Zeitbereich → Multiplikation mit Fensterfunktion ○—● Faltung mit zur Fensterfunktion zugehörigem Spektrum

Dirichlet-Kern

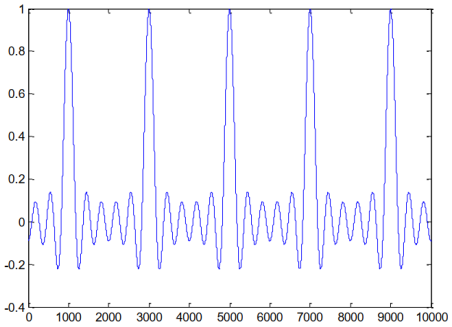
abgetastete Rechteckfunktion ○—● Dirichlet-Kern

Definition der Dirichlet-Kerns der Länge $N + 1$:

$$D(x) = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} e^{jnx} = \frac{\sin\left(\left(\frac{N+1}{2}\right)x\right)}{\sin\left(\frac{x}{2}\right)}$$

$x = 2\pi f T_a$

Bsp: 11:



Eigenschaften:

Hauptwert hat Höhe $N + 1$
Nullstellen, bei $f = \frac{f_a}{N+1} \cdot k$ für $k \in \mathbf{N}$
Hauptwert periodisch mit f_a

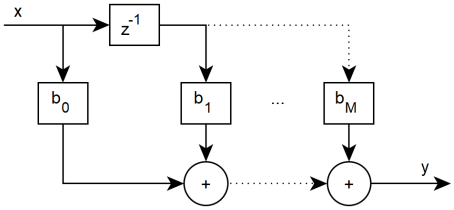
z-Transformation

$$z = e^{j2\pi \frac{f}{f_a}}$$

$$H(f) = H(z)\Big|_{z=e^{j2\pi \frac{f}{f_a}}}$$

Reiner FIR-Filter

kanonische Form

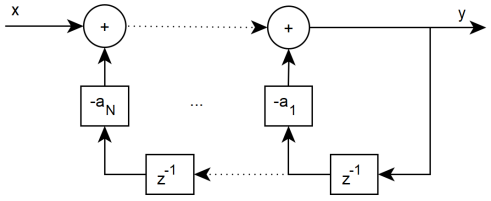


$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 \cdot z^{-1} + ... + b_M \cdot z^{-M}$$

$$y(t) = b_0 \cdot x(t) + b_1 \cdot x(t-1) + ... + b_M \cdot x(t-M)$$

Reiner IIR-Filter

kanonische Form



$$H(z) = \frac{1}{1 + a_1 \cdot z^{-1} + ... + a_N \cdot z^{-N}}$$

$$y(t) = x(t) - a_1 \cdot y(t-1) - ... - a_n \cdot y(t-N)$$

E. Wahrscheinlichkeitstheorie

E.1. Satz von Bayes

$$P(A|B) \cdot P(B) = P(B|A) \cdot P(A) = P(A,B)$$

$$P(A|B) = \frac{P(B|A)}{P(B)} \cdot P(A)$$

E.2. Kombinatorik

E.2.1. Permutation

Ohne Wiederholung

Anordung aller möglicher, unterscheidbarer Ereignisse, kein Ereignisse tritt doppelt auf

$$M = N!$$

Beispiel: In einem Glas sind 5 verschiedenfarbige Bonbons. Wie viele Möglichkeiten gibt es, alle aus dem Glas zu nehmen? (ohne Zurücklegen)

$$\hookrightarrow M = N! = 5! = 120$$

Mit Wiederholung

Anordung aller möglicher Ereignisse, von welchen manche nicht unterscheidbar sind

$$M = \frac{N!}{K_1! \cdot ... \cdot K_s!}$$

K_x: Gibt an, wie viele ununterscheidbare Ereignisse es pro Ereignis gibt

Beispiel: In einem Glas liegen 2 blaue, 2 rote und ein grünes Bonbon. Wie viele Möglichkeiten gibt es alle aus dem Glas zu nehmen? (Ohne Zurücklegen)

$$\hookrightarrow \frac{5!}{2! \cdot 2! \cdot 1!} = 30$$

N: Anzahl der möglichen Ereignisse
M: Anzahl der Permutationen

E.2.2. Variation

Ohne Wiederholung

Zusammenstellung von K Ereignissen; Reihenfolge relevant

$$M = \frac{N!}{(N-K)!}$$

Beispiel: In einem Glas liegen 5 verschiedenfarbige Bonbons. Wie viele Möglichkeiten gibt es, 3 Bonbons herauszunehmen? (Ohne Zurücklegen)

$$\hookrightarrow M = \frac{5!}{(5-2)!} = \frac{5!}{2!} = 60$$

Mit Wiederholung

$$M = N^K$$

Beispiel: In einem Glas liegen 5 verschiedenfarbige Bonbons. Wie viele Möglichkeiten gibt es, 3 Bonbons herauszunehmen? (Mit Zurücklegen)

$$\hookrightarrow M = 5^3 = 125$$

M: Anzahl der Variationen
N: Anzahl der möglichen Ereignisse
K: Anzahl der zusammengestellten Ereignisse (Ordnung)

E.2.3. Kombination

Wie Variation, Reihenfolge aber irrelevant (vgl. Lottoziehung)

Ohne Wiederholung

$$M = \binom{N}{K} = \frac{N!}{K! \cdot (N-K)!}$$

Mit Wiederholung

$$M = \binom{N+K-1}{K} = \frac{(N+K-1)!}{K! \cdot (N-1)!}$$

M: Anzahl der Kombinationen
N: Anzahl der möglichen Ereignisse
K: Anzahl der zusammengestellten Ereignisse (Ordnung)