Zusammenfassung Stoff des Moduls IcTh

Arbeitsversion 0.3

Autor: Michael Klenk mklenk@hsr.ch

Beispiele sind so formatiert

Wahrscheinlichkeit

Wahrscheinlichkeit allgemein

Die Wahrscheinlichkeit des eintreten eines Falles liegt zwischen 0 und 1.
Tritt ein Ereignis nie ein ist die Wahrscheinlichkeit 0.

$$\text{Warschenlichkeit } p = \left(\frac{\text{gültiger Fall}}{\text{alle Fälle}}\right)$$

Bedingte / unbedingte Ereignisse

Treten mehrere verschiedene Ereignisse auf, gibt es 2 Fälle. Entweder hat das Auftreten des 1. Ereignisses einen Einfluss auf das nächste Ereignis oder nicht.

Bsp. Wir nehmen blaue und rote Kugeln aus einer Urne:

Wahrscheinlichkeit dass E1 nach E2 eintritt mit Einfluss

$$p\{E1E2\} = p\{E1\} \bullet p\{E2 \mid E1\}$$

Wir legen die Entnommenen Kugeln nicht in die Urne zurück

Ohne Einfluss von E1

$$p\{E1E2\} = p\{E1\} \bullet p\{E2\}$$

Wir legen die gezogenen Kugeln zurück in die Urne

entweder/oder ausschliessende Ereignisse

Wir ziehen Karten aus einem Kartenspiel

entweder/oder

$$p\{E1 + E2\} = p\{E1\} + p\{E2\} - p\{E1E2\}$$

Wahrscheinlichkeit für König und Kreuz

Ausschliessendes Ereignis

$$p\{E1E2\} = 0$$

$$p\{E1 + E2\} = p\{E1\} + p\{E2\}$$

Wahrscheinlichkeit für 1 Ass und 1 König

Information

Information kann dann Entstehen wenn beim Sender (Quelle) und beim Empfänger (Senke) folgende Bedingungen erfüllt sind:

- Gleicher Zeichenvorrat
- Gleiche Darstellung (Syntax)
- Gleiche Bedeutung (Semantik)

Wann enthält eine Nachricht Information?

Nachricht	redundant	nicht redundant
Irrelevant	Zeichenvorrat verschieden bei Quelle und Senke	
Relevant	vorhersagbar	Information

Redundanz wird gebraucht um Fehlerkorrektur zu ermöglichen.

Entropie

Entscheidungsgehalt eines Zeichens bezeichnet die Anzahl Entscheidungen (Kreuzungen) welche auf einem Binären Entscheidungsbaum (immer nur 2 Abzweigungen) nötig sind um das Zeichen eindeutig zu bestimmen.

$H_0 = lb(\text{anz möglicher Zeichen})$ in Bit

Informationsgehalt eines Zeichens

$$I(x_k) = lb\left(\frac{1}{p(x_k)}\right)[bit]$$

$p(x_k)$ Warscheinlichkeit des Zeichens

Die Entropie einer Quelle errechnet sich aus der Summe der Informationsgehalte aller möglichen Zeichen mal derer Auftrittswahrscheinlichkeit.

$$H(X) = \sum_{k=1}^{N} p(x_k) \cdot lb\left(\frac{1}{p(x_k)}\right)[bit/Zeichen]$$

$$H(X) = \sum_{k=1}^{N} p(x_k) \cdot I(x_k)$$

Bemerkungen:

- Ist die Auftrittswahrscheinlichkeit aller Zeichen gleich ist die Redundanz = 0 und die die Entropie der Quelle maximal
- Redundanz: $R_Q = H_0 - H(X)$

Binärcodierung

Diskrete Quellen

Bei den diskreten Quellen werden beliebige Zeichen auf Binäre Codeworte abgebildet. Am günstigsten ist es wenn die Codewortlänge möglichst gering ist.

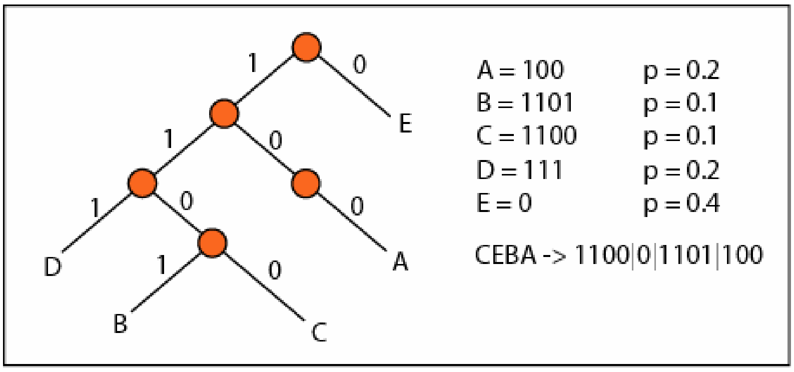
ASCII: Blockcode L= 8 bit
MORSE: L = {1..4} *A → .- in Binär 01 → L = 2 bit*

$$L = \sum_{k=1}^{N} p(x_k) \bullet L(x_k) \left[bit / Zeichen \right]$$

$L(x_k)$ ist immer ganzzahlig keine Kommastellen, aufrunden

Präfixeigenschaft

Beim Morsecode, müssen zwischen den Zeichen ein Trennglied gesendet werden um die verschieden Zeichen trennen zu können. Um dieses Problem zu lösen kann nutzt man kommafreien Code (besitzt Präfixeigenschaft)



Die Zeichen liegen dann immer in den Blättern eines Entscheidungsbaumes.

Für jede Codierung mit Präfixeigenschaft gilt:

$$H(X) \le L$$

Für den Beispiel Code

$$L = 3 \bullet 0.2 + 4 \bullet 0.1 + 4 \bullet 0.1 + 3 \bullet 0.2 + 1 \bullet 0.4$$

$$H(X) = 2.122 \le L = 2.4$$

Berechung der Redundanz eines Codes

$$R_C = L - H(X) \left[bit / Zeichen \right]$$

Für Beispiel Code $R_C = 2.4 - 2.112$
$R_C = 0.278$

Mit Gedächtnis

Die bisherigen Quellen arbeiten ohne Gedächtnis. Allgemein kann nicht von einer gedächtnislosen Quelle ausgegangen werden. Daher benötigen wir für die Berechungen die Verbundentropie der Zeichen

$p(xy) = p(x) \cdot p(y \mid x)$ für eine Quelle $H\langle X, Y \rangle = H\langle X \rangle + H\langle Y \mid X \rangle$

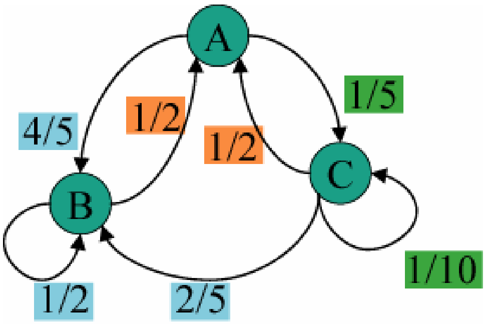
Wichtiges Merkmal:

- Die mittlere Entropie einer Quelle **ohne** Gedächtnis ist immer **grösser oder gleich** der Entropie einer Quelle **mit** Gedächtnis

Bei einer Quelle mit Gedächtnis können wir Voraussagen treffen welches Zeichen als Nächstes kommt, daher sinkt die der Informationsgehalt der Nachricht

Berechnung von Entropie einer Quelle mit Gedächtnis

Eine Quelle mit Gedächtnis kann mithilfe eines Markoff-Diagramms dargestellt werden. Hilfreich ist auch zu wissen, dass die Überlegungen und Berechnungen auch für das Kanalmodell angewendet werden können.



1. Aufstellen der Matrix für die Bedingte Wahrscheinlichkeit. Ablesen der Werte im Markoff-Diagramm.

$p(y \mid x)$	y=			
x=		A	B	C
	A geht über in	0	4/5	1/5
	B geht über in	1/2	1/2	0
	C geht über in	1/2	2/5	1/10

2. Berechnen der Wahrscheinlichkeiten der einzelnen Zeichen

Aufstellen der 4 Gleichungen:

$$1 = p(A) + p(B) + p(C)$$

$$p(A) = p(A) \cdot 0 + p(B) \cdot 1/2 + p(C) \cdot 1/2$$

$$p(B) = p(A) \cdot 4/5 + p(B) \cdot 1/2 + p(C) \cdot 2/5$$

$$p(C) = p(A) \cdot 1/5 + p(B) \cdot 0 + p(C) \cdot 1/10$$

$$p(A) = 1/3$$

$$p(B) = 16/27$$

$$p(C) = 2/27$$

3. Zusammenfassen

$p(yx)$	y=			
x=		A	B	C
	A 1/3 *	0 = **0**	4/5 = **4/15**	1/5 = **1/15**
	B 16/27 *	1/2 = **8/27**	1/2 = **8/27**	0 = **0**
	C 2/27 *	1/2 = **1/27**	2/5 = **4/135**	1/10 = **1/135**

4. Informationsgehalt der Zeichen

$$I(x_k) = lb\left(\frac{1}{p(x_k)}\right)$$ I(A) = 1.585 bit I(B) = 0.755 bit I(C) = 3.755

5. Entropie der Quelle

$$H(X)=\sum_{k=1}^{N} p(x_k)\cdot I(x_k) \quad \frac{1}{3}\cdot 1.585+\frac{16}{27}\cdot 0.755+\frac{2}{27}\cdot 3.755=1.254$$

6. Bedingte Entropie

$$H(Y\,|\,X)=-\sum_{i=1}^{N}\sum_{k=1}^{N} p(x_i)\cdot p(y_k\,|\,x_i)\cdot ld\left(p(x_i)\cdot p(y_k\,|\,x_i)\right)=1.1546$$

Hier wird eine Doppelsumme gebraucht das ist nicht wirklich einfach zu Rechnen und zu verstehen daher hier eine kurze Erklärung:

Man stellt sich das ganze als 2 Matrizen vor. Diese werden nun jedes Element mit jedem multipliziert und Summiert. Die Erste Matrix entsteht bei Schritt 2, die Zweite bei enthält alle Entropien der einzelnen Zeichen aus Schritt 1.

$$\begin{pmatrix} p(A) \\ p(B) \\ p(C) \end{pmatrix} \times \begin{pmatrix} I(a\,|\,a) & I(b\,|\,a) & I(c\,|\,a) \\ I(a\,|\,b) & I(b\,|\,b) & I(c\,|\,b) \\ I(a\,|\,c) & I(b\,|\,c) & I(c\,|\,c) \end{pmatrix}$$

Für alle Faulen hier mein TI-92 Rechnungsprogramm

```
m_entsum(a,b)
Func
Local sa,sb,i,j,tmp
dim(a)→sa
dim(b)→sb
0→tmp

For i,1,sa
      For,j,1,sa
            If b[j+sa*(i-1)]=0 Then
                  next
            Else
                  a[i]*b[j+sa*i-1)]*log(1/b[j+sa*(i-1)]))/log(2))
+tmp→tmp
            EndIf
      EndFor
EndFor
tmp
EndFunc
```

Berechung mit dem Taschenrechner für TI-92 (siehe auch Abschnitt TI92)

Aufruf der Funktion: (Ablesbar direkt aus Schritt 1 + 2)

m_entsum({p(a),p(b),p(c)},{p(a|a),p(a|b),p(a|c),p(b|a),p(b|b),p(b|c),p(c|a),p(c|b),p(c|c)})

m_entsum({1/3,16/27,2/27},{0,1/2,1/2,4/5,1/2,2/5,1/5,0,1/10})
= 1.1545991

7. Verbundentropie

$H\langle X,Y\rangle = H\langle X\rangle + H\langle Y\,|\,X\rangle$ 1.254 + 1.1546 = **2.4086**

Kompression

Das Ziel der Kompression ist die Reduktion der Redundanz und entfernen der Irrelevanz. Man unterscheidet 3 Varianten:

- **Statische Verfahren**
  Huffman-Codierung für bekannte Daten (zb. Deutsche Sprache)
  Blockdaten
- **Adaptive Verfahren**
  Huffman-Codierung mit gemessener Häufigkeitsverteilung
  Blockdaten
- **Dynamische Verfahren**
  LZ77 Lempel, Ziv
  Streamingdaten

Huffman

Die Idee bei der Huffmancodierung liegt darin, dass man einen Code mit möglichst geringer mittlerer Codewortlänge entwickeln möchte.

Man entwickelt einen Binärbaum. Hier beginnt man mit den Blättern und baut ihn auf bis zur Wurzel. (Man erhält dabei einen Kommafreien Code)

Vorgehensweise:

1. Ordnen der vorhandenen Zeichen anhand der Auftrittswahrscheinlichkeit

X	A	B	C	D	E	F
P(x)	0.5	0.15	0.12	0.1	0.08	0.05

2. Immer die 2 Zeichen mit der geringsten Auftrittswahrscheinlichkeit werden Codiert. Der einen Seite wird immer eine 1 der andere eine 0 vorangestellt. Ob mit 0 oder 1 begonnen wird ist egal, es muss einfach Kosequent gleich gemacht werden.

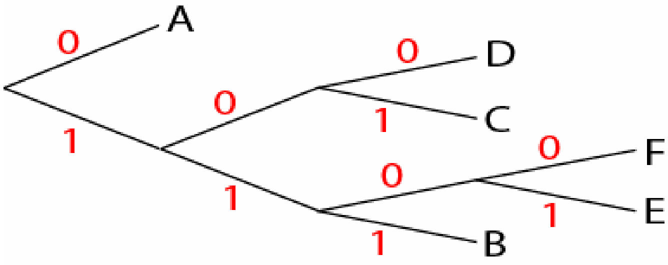
X	A	B	E	F	C	D
Codierung			1	0		
P(x)	0.5	0.15	0.08 + 0.05 = **0.13**		0.12	0.1

X	A	C	D	B	E	F
Codierung		1	0		1	0
P(x)	0.5	0.12 + 0.1 = **0.22**		0.15	0.13	

X	A	B	E	F	C	D
Codierung		1	01	00	1	0
P(x)	0.5	0.15 + 0.13 = **0.28**			0.21	

X	B	E	F	C	D	A
Codierung	11	101	100	01	00	
P(x)	0.28 + 0.22 = **0.5**					0.5

X	B	E	F	C	D	A
Codierung	111	1101	1100	101	100	0
P(x)	0.5 + 0.5 = **1.0**					



Redundanz dieses Codes:

$R_C = L - H(X)\left[bit / Zeichen\right]$ siehe Entropie und Diskrete Quellen

$L = 3 \cdot 0.15 + 4 \cdot 0.08 + 4 \cdot 0.05 + 3 \cdot 0.12 + 3 \cdot 0.1 + 1 \cdot 0.5 = 2.13$

$H(X) = \sum_{k=1}^{N} p(x_k) \bullet I(x_k) = 2.1174$

$R_C = 2.13 - 2.1174 = 0.0126$

Einfluss auf die Redundanz des Codes:

Verändert sich die Auftrittswahrscheinlichkeit der Zeichen, verändert sich auch die Redundanz. Um eine Verbesserung zu realisieren muss der Huffman-Code neu berechnet werden.

Bemerkungen:

Wird bei der Übertragung nur ein Bit falsch übertragen, kann der Rest der Nachricht nicht mehr korrekt Decodiert werden.

Es gibt verschiedene Lösungen die bei der Huffman-Codierung entstehen können. Wichtig ist, dass alle die Codewörter immer in einem Blatt enden.

Link zu einem einfach verständlichen Applet:
http://www.ruhr-uni-bochum.de/ika/ika/lehre/applets/Optimalcodierung/DecoderApplet.html

Lempel-Ziv

Codierungsverfahren für streaming Daten.

Funktionsweise:

1. Die zu Codierende Zeichenfolge wird in Teilfolgen aufgeteilt die alle verschieden sind.
2. Bei den zu übertragenen Zeichen werden die Position im Baum und eventuell neue Zeichenfolgen angeben. (3,1) → (Position 3 im Baum, neues Zeichen 1)
3. Es wird ein Phrasenspeicher (Decoderbaum) aufgebaut, welcher schon übertragene Zeichenfolgen speichert.

Lempel-Ziv LZ77 Binärstreem Codierung / Decodierung

Zeichenfolge zum Codieren

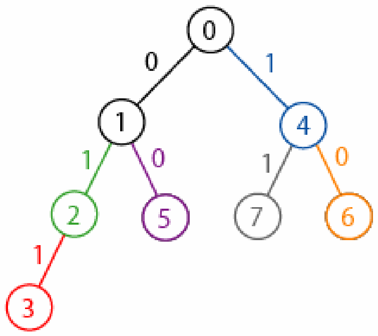
00101110010110

1. Aufteilen in eindeutige Zeichenfolgen

0|01|011|1|00|10|11|0

2. Übertragen

0	(0,0)
01	(1,1)
011	(2,1)
1	(0,1)
00	(1,0)
10	(4,0)
11	(4,1)
0	(1)



3. Decodieren

Bei jedem ankommeden Zeichenpaar, wird der Baum biz zu entsprechenden Knoten verfolgt und die Bits auf dem Weg ausgegeben. Danach wird bei bedarf ein neuer Knoten in den Baum eingefügt.

Bsp. (2,1) -> Verfolgen des Baumes bis Knoten 2. -> Ausgabe der Bits 01 -> Anfügen eines neuen Knoten uns ausgabe des Weges dorthin 1 -> Gesamte Bitfolge ist 011

Voraussetzung für gute Komprimierung ist, dass die Zeichenfolge viele Regelmässigkeiten besitzt.

Probleme:

- Effizientes Suchen und Einfügen in den Baum
- Grösse des Baumes. Der Baum kann nur bis zu einer gewissen Grösse wachsen.

Kanalmodell

Bei jeder Übertragung von Daten über einen Kanal passieren Fehler. Kein Kanal in der Praxis ist fehlerfrei. Daher versuchen wir unseren Kanal in einem Modell zu beschreiben. Das Modell ermöglicht es uns verschiedenste Eigenschaften des Kanals zu berechnen.

Kanalmatrix

Die Kanalmatrix beschreibt die Übergangswahrscheinlichkeit von Zeichen am Eingang des Kanals gegenüber dem Ausgang. $Zeichen(A,B)\begin{bmatrix} p(A\rightarrow A) & p(A\rightarrow B) \\ p(B\rightarrow A) & p(B\rightarrow B) \end{bmatrix}$

Pro Zeile muss die Summe immer 1 sein.

Optimaler Kanal (Einheitsmatrix)	Vollständig gestörter Kanal keine Übertragung möglich	Teilweise gestörter Kanal Praxis
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$	$\begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$

Die Auftrittswahrscheinlichkeit der Zeichen verändert sich mit der Kanalmatrix

$$\begin{matrix} p(A)=0.3 \\ p(B)=0.1 \\ p(C)=0.6 \end{matrix} \quad \text{Kanalmatrix} \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.2 & 0.2 & 0.6 \end{bmatrix} \rightarrow \begin{bmatrix} 0.3\cdot0.8+0.1\cdot0.2+0.6\cdot0.2 \\ 0.3\cdot0.1+0.1\cdot0.5+0.6\cdot0.2 \\ 0.3\cdot0.1+0.1\cdot0.3+0.6\cdot0.6 \end{bmatrix} = \begin{matrix} p(A')=0.38 \\ p(B')=0.2 \\ p(C')=0.42 \end{matrix}$$

Maximum Likelihood Verfahren

Bei diesem verfahren wird mein Empfänger anhand der Kanalmatrix das am wahrscheinlichten angelegene Zeichen ausgewählt. Ist jedes Zeichen gleich wahrscheinlich wird ausgewählt.

Die Entscheidungsleistung kann noch gesteigert werden, wenn die Auftrittswahrscheinlichkeit der Zeichen an der Quelle berücksichtigt werden.

$$\text{Kanalmatrix } p(Y\,|\,X) = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.6 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}$$

Man sieht dass im Falle des 2. Zeichens keine Aussage getroffen werden kann welches Zeichen angelegen haben mag.

Restfehler $P_F = 1 - P_R$

$$P_R = \sum_{i=1}^{n} p(x_i)\cdot p(y_i\,|\,x_i)$$

$$1-\left(\frac{1}{3}\cdot0.6+\frac{1}{3}\cdot0.2+\frac{1}{3}\cdot0.6\right)=0.53$$

Transinformation

Kanal Codierung

Blockcodes

Zyklische Codes

Faltungscodes

Encoder

Decoder

Signale

Darstellung

Zeit und Wertdiskret

Tiefpass / Hochpass / Bandpass

Kompression

Quantisierung

Leitungscodierung

Manchester

AMI Code

GFSK / DPSK

Rechner TI 92

Wer einen Texas Rechner TI 92/89 oder ähnlich besitzt kann folgende Befehle mit hilfe gut gebrauchen.

Listen

Eine neue Liste wird wie Folgt definiert:

{[x1].[x2],[x3]….,[xn]}→[var_name]   → Ist auf dem TI92 die Taste STO >

{1,2,3,4,5] →*myVar*

Summen

Wichtig ist bei Listen wird mit dem Index 1 begonnen nicht wie sonst üblich bei 0

$\sum([term],[zaehl\_\mathrm{var}],[start],[ende])$

$\sum(2\wedge n,n,1,8)=126$ $\sum$ Ist auf dem TI92 [2nd] + 4

$\sum(myVar[n],n,1,\dim(myVar))=15$