

Vorbereitende Theorie und Etablierung eines Algorithmus zur Bestimmung einer optimalen Strategie

Torsten Grote

Universität Potsdam - Institut für Informatik

25.04.2005

Gliederung

1 Ungleichung von Kraft

Gliederung

- 1 Ungleichung von Kraft
- 2 noiseless-coding-theorem

Gliederung

- 1 Ungleichung von Kraft
- 2 noiseless-coding-theorem
- 3 Der Algorithmus von Huffman
 - mathematisch
 - in Worten
 - am Beispiel

Gliederung

- 1 Ungleichung von Kraft
- 2 noiseless-coding-theorem
- 3 Der Algorithmus von Huffman
 - mathematisch
 - in Worten
 - am Beispiel
- 4 Optimale Strategien bei Gleichverteilung auf dem Suchbereich

Gliederung

- 1 Ungleichung von Kraft
- 2 noiseless-coding-theorem
- 3 Der Algorithmus von Huffman
 - mathematisch
 - in Worten
 - am Beispiel
- 4 Optimale Strategien bei Gleichverteilung auf dem Suchbereich
- 5 Quellenverzeichnis

Ungleichung von Kraft

Präfixcodes

- Präfixcodes lassen sich als Binärbäume darstellen.
- Sequentielle Suchstrategien und Präfixcodes sind äquivalent.
- Im Folgenden wird der Einfachheit halber nur von Präfixcodes die Rede sein.

Ungleichung von Kraft

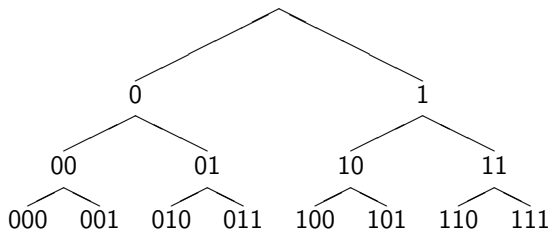
Präfixcodes

- Präfixcodes lassen sich als Binärbäume darstellen.
- Sequentielle Suchstrategien und Präfixcodes sind äquivalent.
- Im Folgenden wird der Einfachheit halber nur von Präfixcodes die Rede sein.

Problemstellung

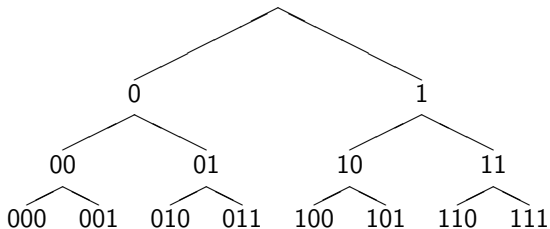
- Zu welchen Codewortlängen $L(1), \dots, L(n)$ existiert ein Präfixcode?
- Lösung anhand von Binärbäumen

Ungleichung von Kraft



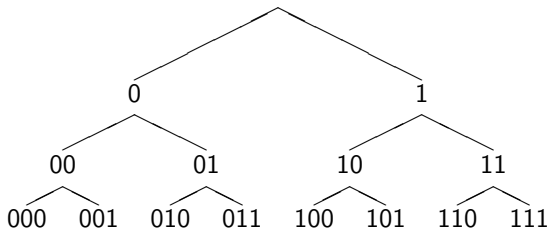
- Wenn das längste Codewort eines Präfixcodes die Länge L_{max} hat, kommt jedes Codewort im Binärbaum T_{max} der Länge L_{max} vor.

Ungleichung von Kraft



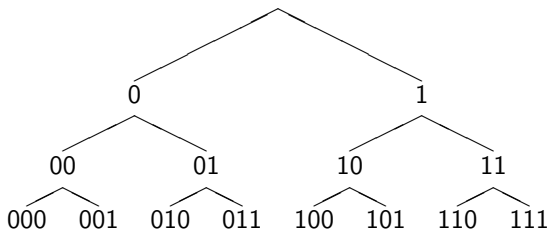
- Wenn das längste Codewort eines Präfixcodes die Länge L_{max} hat, kommt jedes Codewort im Binärbaum T_{max} der Länge L_{max} vor.
- Codewort der Länge L hat in T_{max} genau $2^{L_{max}-L}$ Blätter als Nachfolger. Bsp: '01' hat $2 = 2^{3-2}$ Nachfolger.

Ungleichung von Kraft



- Wenn das längste Codewort eines Präfixcodes die Länge L_{max} hat, kommt jedes Codewort im Binärbaum T_{max} der Länge L_{max} vor.
- Codewort der Länge L hat in T_{max} genau $2^{L_{max}-L}$ Blätter als Nachfolger. Bsp: '01' hat $2 = 2^{3-2}$ Nachfolger.
- T_{max} hat $2^{L_{max}}$ Blätter und es gilt:
$$\sum_{1 \leq j \leq n} 2^{L_{max}-L(j)} \leq 2^{L_{max}}$$

Ungleichung von Kraft



- Wenn das längste Codewort eines Präfixcodes die Länge L_{max} hat, kommt jedes Codewort im Binärbaum T_{max} der Länge L_{max} vor.
- Codewort der Länge L hat in T_{max} genau $2^{L_{max}-L}$ Blätter als Nachfolger. Bsp: '01' hat $2 = 2^{3-2}$ Nachfolger.
- T_{max} hat $2^{L_{max}}$ Blätter und es gilt:
$$\sum_{1 \leq j \leq n} 2^{L_{max}-L(j)} \leq 2^{L_{max}}$$
- folglich gilt
$$2^{L_{max}} \cdot \sum_{1 \leq j \leq n} \frac{1}{2^{L(j)}} \leq 2^{L_{max}} \quad \text{und} \quad \sum_{1 \leq j \leq n} 2^{-L(j)} \leq 1$$

Ungleichung von Kraft

Die Ungleichung von Kraft

Es gibt genau dann einen Präfixcode im Alphabet $\{0, 1\}$ mit Codewortlängen $L(1), \dots, L(n)$, wenn $\sum_{1 \leq j \leq n} 2^{-L(j)} \leq 1$ ist.

- Die Aussage gilt auch bei einem k -elementigen Alphabet, wenn man die 2 durch k ersetzt.

Ungleichung von Kraft

Die Ungleichung von Kraft

Es gibt genau dann einen Präfixcode im Alphabet $\{0, 1\}$ mit Codewortlängen $L(1), \dots, L(n)$, wenn $\sum_{1 \leq j \leq n} 2^{-L(j)} \leq 1$ ist.

- Die Aussage gilt auch bei einem k -elementigen Alphabet, wenn man die 2 durch k ersetzt.

Entropie

- Entropie ist Maß für den Informationsgehalt einer Nachricht
- hier als Maß für die Unsicherheit über den Wert einer Zufallsvariablen mit der Verteilung p interpretiert

Die Entropie der Wahrscheinlichkeitsverteilungen $p = (p(1), \dots, p(n))$ ist

$$H(p) := - \sum_{1 \leq i \leq n} p(i) \cdot \log_2 p(i).$$

noiseless-coding-theorem

Fragestellung

Gibt es Schranken für die erwartete Codewortlänge eines optimalen Präfixcodes?

Sei für $i \in \{1, \dots, n\}$ $p(i) > 0$ und $L_{min}(p)$ die erwartete Codewortlänge eines optimalen Präfixcodes zur a-priori Verteilung p . Dann ist

$$H(p) \leq L_{min}(p) < H(p) + 1$$

noiseless-coding-theorem

Fragestellung

Gibt es Schranken für die erwartete Codewortlänge eines optimalen Präfixcodes?

Sei für $i \in \{1, \dots, n\}$ $p(i) > 0$ und $L_{min}(p)$ die erwartete Codewortlänge eines optimalen Präfixcodes zur a-priori Verteilung p . Dann ist

$$H(p) \leq L_{min}(p) < H(p) + 1$$

- Die erwartete Codewortlänge bzw. die erwartete Suchdauer einer optimalen sequentiellen Suchstrategie ist also mindestens $H(p)$ und kleiner als $H(p) + 1$.
- Je größer die Entropie der Wahrscheinlichkeitsverteilung, desto länger dauert im Schnitt eine optimale Suche.
- Das noiseless-coding-theorem gilt auch bei einem k-elementigen Alphabet, wenn man die Entropie mit

$$H(p) := - \sum_{1 \leq i \leq n} p(i) \cdot \log_k p(i) \text{ berechnet.}$$

Der Algorithmus von Huffman

Problemstellung

Wie kann man optimale Präfixcodes bzw. optimale Suchstrategien ermitteln?

Der Algorithmus von Huffman

Problemstellung

Wie kann man optimale Präfixcodes bzw. optimale Suchstrategien ermitteln?

Lösung

- 1952 fand David A. Huffman einen Algorithmus, der beweisbar immer einen optimalen Baum liefert.
- Der Baum repräsentiert den Präfixcode und seine Blätter die Nachrichten bzw. die Codewörter.
- Die Häufigkeitsverteilung der verschiedenen Nachrichten muss allerdings bekannt sein.

Der Algorithmus von Huffman

Sei $p = (p(1), \dots, p(n))$ eine a-priori-Verteilung für n Nachrichten, wobei $p(1) \geq \dots \geq p(n)$ ist.

Sei $p' = (p(1), \dots, p(n-2), p(n-1) + p(n))$ und sei

$c' = (c'(1), \dots, c'(n-1))$ ein optimaler Präfixcode zu p' . Sei für $j \in \{1, \dots, n-2\}$ $c(j) := c'(j)$ und sei $c(n-1)$ bzw. $c(n)$ die

Verlängerung von $c'(n-1)$ um eine 0 bzw. 1.

Dann ist c ein optimaler Code zu p und

$$L_{\min}(p) - L_{\min}(p') = E(c) - E(c') = p(n-1) + p(n).$$

Der Algorithmus von Huffman

- 1 Sei N_i die i -te Nachricht. Ordne $p(1), \dots, p(n)$, so dass $p(i_1) \geq \dots \geq p(i_n)$ ist.

Der Algorithmus von Huffman

- 1 Sei N_i die i -te Nachricht. Ordne $p(1), \dots, p(n)$, so dass $p(i_1) \geq \dots \geq p(i_n)$ ist.
- 2 Falls $i_n \geq 2$, fasse die Nachrichten N_i und N_j mit den kleinsten Wahrscheinlichkeiten zu einer neuen Nachricht $N_{(i,j)}$ mit $p((i,j)) := p(i) + p(j)$ zusammen. Entferne nun N_i mit $p(i)$ sowie N_j mit $p(j)$ und füge dafür $N_{(i,j)}$ mit $p((i,j))$ ein. Wiederhole nun solange, bis nur noch zwei Nachrichten vorliegen, denen die Codeworte 0 und 1 zugeordnet werden.

Der Algorithmus von Huffman

- 1 Sei N_i die i -te Nachricht. Ordne $p(1), \dots, p(n)$, so dass $p(i_1) \geq \dots \geq p(i_n)$ ist.
- 2 Falls $i_n \geq 2$, fasse die Nachrichten N_i und N_j mit den kleinsten Wahrscheinlichkeiten zu einer neuen Nachricht $N_{(i,j)}$ mit $p((i,j)) := p(i) + p(j)$ zusammen. Entferne nun N_i mit $p(i)$ sowie N_j mit $p(j)$ und füge dafür $N_{(i,j)}$ mit $p((i,j))$ ein. Wiederhole nun solange, bis nur noch zwei Nachrichten vorliegen, denen die Codeworte 0 und 1 zugeordnet werden.
- 3 Durchlaufe 2. in umgekehrter Reihenfolge. Falls N_i und N_j zu $N_{(i,j)}$ zusammengefasst wurden und $c((i,j))$ das Codewort dafür ist, dann seien $c(i)$ und $c(j)$ die Verlängerungen von $c((i,j))$ um 0 und 1.

Der Algorithmus von Huffman

- 1 Sei N_i die i -te Nachricht. Ordne $p(1), \dots, p(n)$, so dass $p(i_1) \geq \dots \geq p(i_n)$ ist.
- 2 Falls $i_n \geq 2$, fasse die Nachrichten N_i und N_j mit den kleinsten Wahrscheinlichkeiten zu einer neuen Nachricht $N_{(i,j)}$ mit $p((i,j)) := p(i) + p(j)$ zusammen. Entferne nun N_i mit $p(i)$ sowie N_j mit $p(j)$ und füge dafür $N_{(i,j)}$ mit $p((i,j))$ ein. Wiederhole nun solange, bis nur noch zwei Nachrichten vorliegen, denen die Codeworte 0 und 1 zugeordnet werden.
- 3 Durchlaufe 2. in umgekehrter Reihenfolge. Falls N_i und N_j zu $N_{(i,j)}$ zusammengefasst wurden und $c((i,j))$ das Codewort dafür ist, dann seien $c(i)$ und $c(j)$ die Verlängerungen von $c((i,j))$ um 0 und 1.
- 4 $c = (c(1), \dots, c(n))$ ist ein optimaler Präfixcode für N_1, \dots, N_n .

Der Algorithmus von Huffman

Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5	0,40
-------	------

N_2	0,20
-------	------

N_1	0,15
-------	------

N_4	0,15
-------	------

N_3	0,10
-------	------

Der Algorithmus von Huffman

Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5	0,40	N_5	0,40
N_2	0,20	$N_{(4,3)}$	0,25
N_1	0,15	N_2	0,20
N_4	0,15	N_1	0,15
N_3	0,10		

Der Algorithmus von Huffman

Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5	0,40	N_5	0,40	N_5	0,40
N_2	0,20	$N_{(4,3)}$	0,25	$N_{(2,1)}$	0,35
N_1	0,15	N_2	0,20	$N_{(4,3)}$	0,25
N_4	0,15	N_1	0,15		
N_3	0,10				

Der Algorithmus von Huffman

Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5	0,40	N_5	0,40	N_5	0,40	$N_{((2,1),(4,3))}$	0,60
N_2	0,20	$N_{(4,3)}$	0,25	$N_{(2,1)}$	0,35	N_5	0,40
N_1	0,15	N_2	0,20	$N_{(4,3)}$	0,25		
N_4	0,15	N_1	0,15				
N_3	0,10						

Der Algorithmus von Huffman

Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5 0,40	N_5 0,40	N_5 0,40	$N_{((2,1),(4,3))}$ 0,60
N_2 0,20	$N_{(4,3)}$ 0,25	$N_{(2,1)}$ 0,35	N_5 0,40
N_1 0,15	N_2 0,20	$N_{(4,3)}$ 0,25	
N_4 0,15	N_1 0,15		
N_3 0,10			

Nun werden rückwärts die Codewörter vergeben.

$N_{((2,1),(4,3))}$	0
N_5	1

Der Algorithmus von Huffman

Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5 0,40	N_5 0,40	N_5 0,40	$N_{((2,1),(4,3))}$ 0,60
N_2 0,20	$N_{(4,3)}$ 0,25	$N_{(2,1)}$ 0,35	N_5 0,40
N_1 0,15	N_2 0,20	$N_{(4,3)}$ 0,25	
N_4 0,15	N_1 0,15		
N_3 0,10			

Nun werden rückwärts die Codewörter vergeben.

$N_{((2,1),(4,3))}$ 0	N_5 1
N_5 1	$N_{(2,1)}$ 00
	$N_{(4,3)}$ 01

Der Algorithmus von Huffman

Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5 0,40	N_5 0,40	N_5 0,40	$N_{((2,1),(4,3))}$ 0,60
N_2 0,20	$N_{(4,3)}$ 0,25	$N_{(2,1)}$ 0,35	N_5 0,40
N_1 0,15	N_2 0,20	$N_{(4,3)}$ 0,25	
N_4 0,15	N_1 0,15		
N_3 0,10			

Nun werden rückwärts die Codewörter vergeben.

$N_{((2,1),(4,3))}$ 0	N_5 1	N_5 1
N_5 1	$N_{(2,1)}$ 00	$N_{(4,3)}$ 01
	$N_{(4,3)}$ 01	N_2 000
		N_1 001

Der Algorithmus von Huffman

Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5 0,40	N_5 0,40	N_5 0,40	$N_{((2,1),(4,3))}$ 0,60
N_2 0,20	$N_{(4,3)}$ 0,25	$N_{(2,1)}$ 0,35	N_5 0,40
N_1 0,15	N_2 0,20	$N_{(4,3)}$ 0,25	
N_4 0,15	N_1 0,15		
N_3 0,10			

Nun werden rückwärts die Codewörter vergeben.

$N_{((2,1),(4,3))}$ 0	N_5 1	N_5 1	N_5 1
N_5 1	$N_{(2,1)}$ 00	$N_{(4,3)}$ 01	N_2 000
	$N_{(4,3)}$ 01	N_2 000	N_1 001
		N_1 001	N_4 010
			N_3 011

Der Algorithmus von Huffman

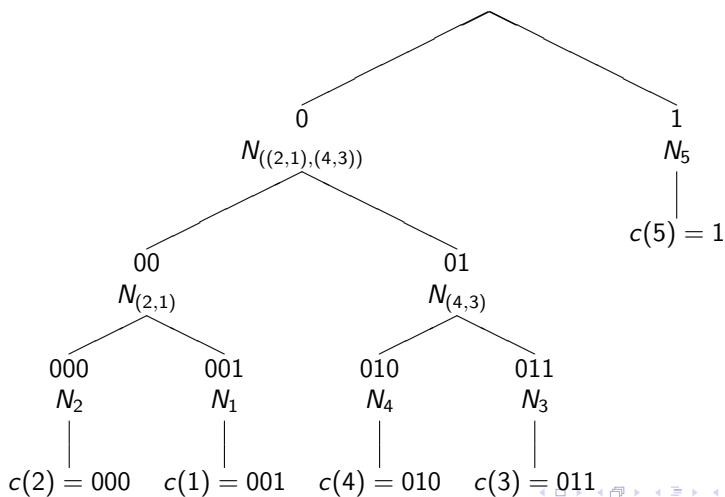
Für 5 Nachrichten mit den Wahrscheinlichkeiten $p(1) = 0,15$, $p(2) = 0,20$, $p(3) = 0,10$, $p(4) = 0,15$ und $p(5) = 0,40$ soll ein optimaler Präfixcode gefunden werden.

N_5 0,40	N_5 0,40	N_5 0,40	$N_{((2,1),(4,3))}$ 0,60
N_2 0,20	$N_{(4,3)}$ 0,25	$N_{(2,1)}$ 0,35	N_5 0,40
N_1 0,15	N_2 0,20	$N_{(4,3)}$ 0,25	
N_4 0,15	N_1 0,15		
N_3 0,10			

Nun werden rückwärts die Codewörter vergeben.

$N_{((2,1),(4,3))}$ 0	N_5 1	N_5 1	N_5 1	$=: c(5)$
N_5 1	$N_{(2,1)}$ 00	$N_{(4,3)}$ 01	N_2 000	$=: c(2)$
	$N_{(4,3)}$ 01	N_2 000	N_1 001	$=: c(1)$
		N_1 001	N_4 010	$=: c(4)$
			N_3 011	$=: c(3)$

Der Algorithmus von Huffman



Der Algorithmus von Huffman

$$\begin{aligned} H(p) &= H((p(1), p(2), p(3), p(4), p(5))) = - \sum_{1 \leq i \leq n} p(i) \cdot \log_2 p(i) \\ &= p(1) \cdot \log_2 p(1) + \dots + p(5) \cdot \log_2 p(5) \\ &\approx 2,146 \end{aligned}$$

$$H(p) \approx 2,146 \leq L_{\min}(p) < 3,146 \approx H(p) + 1$$

Der Algorithmus von Huffman

$$\begin{aligned}H(p) &= H((p(1), p(2), p(3), p(4), p(5))) = - \sum_{1 \leq i \leq n} p(i) \cdot \log_2 p(i) \\ &= p(1) \cdot \log_2 p(1) + \dots + p(5) \cdot \log_2 p(5) \\ &\approx 2,146\end{aligned}$$

$$H(p) \approx 2,146 \leq L_{\min}(p) < 3,146 \approx H(p) + 1$$

$$\begin{aligned}E(c) &= E((c(1), c(2), c(3), c(4), c(5))) = \sum_{1 \leq i \leq n} p(i) \cdot L(i) \\ &= p(1) \cdot L(1) + p(2) \cdot L(2) + p(3) \cdot L(3) + p(4) \cdot L(4) + p(5) \cdot L(5) \\ &= 0,15 \cdot 3 + 0,20 \cdot 3 + 0,10 \cdot 3 + 0,15 \cdot 3 + 0,40 \cdot 1 \\ &= 2,2\end{aligned}$$

Optimale Strategien bei Gleichverteilung auf dem Suchbereich

Problemstellung

Wie kann die erwartete Suchdauer bei Gleichverteilung bestimmt werden?

Optimale Strategien bei Gleichverteilung auf dem Suchbereich

Problemstellung

Wie kann die erwartete Suchdauer bei Gleichverteilung bestimmt werden?

Längendifferenz

Sei c ein für p_n optimaler Präfixcode mit den Codewortlängen $L(1) \leq \dots \leq L(n)$. Dann ist $L(n) - L(1) \leq 1$.

Optimale Strategien bei Gleichverteilung auf dem Suchbereich

Problemstellung

Wie kann die erwartete Suchdauer bei Gleichverteilung bestimmt werden?

Längendifferenz

Sei c ein für p_n optimaler Präfixcode mit den Codewortlängen $L(1) \leq \dots \leq L(n)$. Dann ist $L(n) - L(1) \leq 1$.

Codewortlänge

$L_{\min}(p_n) = \lceil \log_2 n \rceil - \frac{1}{n} 2^{\lceil \log_2 n \rceil} + 1$. Wenn c ein für p_n optimaler Präfixcode ist, haben $a(n) = 2^{\lceil \log_2 n \rceil} - n$ Codeworte die Länge $\lceil \log_2 n \rceil - 1$ und $b(n) = 2n - 2^{\lceil \log_2 n \rceil}$ Codeworte die Länge $\lceil \log_2 n \rceil$.

Quellenverzeichnis



Rudolf Ahlswede und Ingo Wegener:
Suchprobleme
Stuttgart, 1979, S. 25–37.



Wikipedia. Die freie Enzyklopädie:
Stichwort: Shannon-Fano-Kodierung
Stand: 16.04.2005

Fragen?