ETH Zürich
Dept. Informatik

Prof. Markus Gross
WS 06/07

Informationstheorie
Übung 8

Ausgabe: 18. Dezember 2006
Abgabe: 08. Januar 2006

8.1 Arithmetische Codierung

a) Betrachten Sie folgendes Wahrscheinlichkeitsmodell für eine Quelle mit drei Buchstaben a, b und c:

Buchstabe	Wahrscheinlichkeit
a	50%
b	30%
c	20%

Bestimmen Sie einen reellwertigen Stellvertreter des Wortes *aacbca* durch arithmetische Codierung.

b) Finden Sie das Wort der Länge 10, das 0.63215699 als Stellvertreter hat. Schreiben Sie dazu ein kurzes Programm (in Matlab, Maple, oder einer Sprache Ihrer Wahl), das diese Decodierung durchführt.

c) Sei $x = x_1 x_2 ... x_m$ der zu codierende String und seien $u^{(m)}$ und $o^{(m)}$ die Unter- und Obergrenze des dazugehörigen Intervalls. In dieser Aufgabe wollen wir den Mittelpunkt des resultierenden Intervalls als Stellvertreter eines Wortes wählen. Die reellwertige Repräsentation ist folglich $\overline{T}(x_1 x_2 ... x_m) = \frac{o^{(m)} - u^{(m)}}{2}$.

Die binäre Darstellung der reellwertigen Repräsentation kann im Allgemeinen beliebig lang, bzw. unendlich sein. Um unsere reellwertige Repräsentation binär darzustellen, betrachten wir deswegen nur eine beschränkte Anzahl Bits nach dem Komma. Wir nehmen für diese Aufgabe die $l(x) = \lceil \log \frac{1}{P(x)} \rceil + 1$ ersten Bits, $\lfloor \overline{T}(x) \rfloor_{l(x)}$, wobei $P(x)$ die Wahrscheinlichkeit des Wortes x ist.

Zeigen Sie, dass diese gekürzte Repräsentation korrekt ist, indem Sie zeigen, dass sich $\lfloor \overline{T}(x) \rfloor_{l(x)}$ im Intervall $\left[u^{(m)}, o^{(m)} \right)$ des Strings x befindet.

8.2 Das „Perpetuum mobile" der Datenkompression

a) Ein Erfinder möchte Ihrer Softwarefirma ein völlig revolutionäres Datenkompressionsverfahren verkaufen. Mit dem Verfahren soll es angeblich möglich sein, auf einer Festplatte mit 100 GB Kapazität 110 GB abzuspeichern. Was ist davon zu halten?

b) Sie geben dem Erfinder 110 GB zufällig generierte Bits als Testdaten. Am nächsten Tag präsentiert er Ihnen die neueste Version des Programms, welche tatsächlich die

110 GB auf 100 GB komprimiert; zudem können die ursprünglichen Daten einwandfrei rekonstruiert werden.

Wie erklären Sie sich dies? Widerspricht dieses Beispiel der Aussage, dass die mittlere Codewortlänge beim Codieren einer Zufallsvariable X mittels eines (binären) Codes immer grösser als $H(X)$ ist?

8.3 Codierung der ganzen Zahlen

In der Vorlesung wurde eine Methode für die präfixfreie Codierung ganzer Zahlen vorgestellt.

a) Stellen Sie eine Tabelle der Codes C_1 und C_2 für die Zahlen von 1 bis 10 auf. Berechnen Sie auch $C_1(2^{19})$ und $C_2(2^{19})$.

b) Beim Code C_2 wird das erste Symbol „1" des binären Codes B weggestrichen. Warum kann man als ersten Teil des Codewortes nicht einfach die um 1 reduzierte Länge codieren (mit C_1)?

c) Modifizieren Sie den Code C_2 so, dass er präfixfrei bleibt, aber für alle Wahrscheinlichkeitsverteilungen mit $P(1) < P(2) + P(3)$ besser ist als C_2.

8.4 Intervalllängen-Codierung

a) Es soll die Sequenz $s = \langle a, b, c, a, c, c, b, b \rangle$ über dem Alphabet $\mathcal{X} = \{a, b, c\}$ intervalllängencodiert werden. Geben Sie die resultierende Zahlensequenz an. Die ersten drei Zeichen dienen hier als Initialisierung und sollen nicht codiert werden.

b) Geben Sie eine Implementierung für die Intervalllängencodierung an.

c) In welche Zahl würden a, b oder c codiert, wenn sie das nächste Zeichen nach der Sequenz s wäre? Liesse sich diese Codierung verbessern, d.h. liesse sich jedes Zeichen in eine kleinere (oder gleichgrosse) Zahl codieren? Wie lautet die allgemeine Regel für ihre neue Codierung?

d) Geben Sie auch für die modifizierte Codierung eine Implementierung an. Vergleichen Sie diesen Code mit dem aus Aufgabe b) bezüglich der Effizienz.