Komprimierungstechniken

Sabine Oberleiter

28.05.2014

SS 14

Inhaltsverzeichnis

1 Einleitung **2**

2 Hauptteil **2**

2.1 Allgemein 2
2.2 Shannon-Fano-Kodierung 2
2.3 Huffmann-Kodierung 3
2.4 arithmetische Kodierung 3
2.5 Lempel-Ziv-Kodierung 4
2.5.1 LZ77 & LZSS 4
2.5.2 LZ78, LZC & LZW 4

3 Schlussfolgerung **4**

1 Einleitung

Komprimierungstechniken sind für jeden Benutzer eines Computers durchaus von persönlichem Interesse. Beginnend bei der Anforderung eine eigene Fotographie per e-mail zu versenden oder sein eigenes Video auf eine DVD zu brennen. Immer wieder erreicht man den Punkt, wo die Datenmenge viel zu groß im Vergleich zu dem zur Verfügung stehenden Speicherplatz oder der Leitungsbandbreite des Übertragungsweges ist. Dies soll als Überblick für die verschiedensten Möglichkeiten der Komprimierungsverfahren dienen und als Hilfestellung beim Auswählen des geeigneten Mittels sein.

2 Hauptteil

2.1 Allgemein

Komprimierung oder Kompression beschreibt die Technik, die große Datenmengen durch verschiedene Algorithmen verkleinert. Grundsätzlich gibt es 2 Gruppen der Datenkomprimierungsverfahren:

Verlustbehaftete Kompression (lossy compression): Bei diesem Verfahren wird ein Teil der Information bewußt gelöscht, wobei es sich hier um sogenannte irrelevante Datenreduktion handelt. Dieses Verfahren wird vor allem im Media-Bereich angewendet, da die menschlichen Sinne ohnehin nicht alle verfügbaren Informationen verarbeiten können und es außerhalb der Wahrnehmung bleibt, wenn diese Informationen gelöscht werden.

Verlustlose Kompression (lossless compression): Die dekodierten Daten unterscheiden sich nicht von den Originaldaten, dh dass alle Daten beim Komprimieren erhalten und beim Dekomprimieren wiederhergestellt werden. Die Qualität des Originals wäre in diesem Fall nicht wahrnehmbar beeinflusst, da alle Informationen gleichermaßen relevant sind. Dieses Verfahren wird vor allem im Tabellen- und Textbereich verwendet.

2.2 Shannon-Fano-Kodierung

Die Shannon-Fano-Kodierung sortiert die Zeichen nach Auftrittshäufigkeit, beginnend mit dem am häufigsten auftretenden Zeichen. Diese Bit-Kodierung erfolgt nach folgendem Schema: die meist auftretenden Zeichen bekommen die kleinste Anzahl Bits zugewiesen - bis hin zu den am wenigsten auftretenden Zeichen, welche die umfangreichste Bitkodierung erhalten. Die Shannon-Fano-Kodierung lässt sich in einem Baumdiagramm darstellen und arbeitet sich von oben nach unten durch.

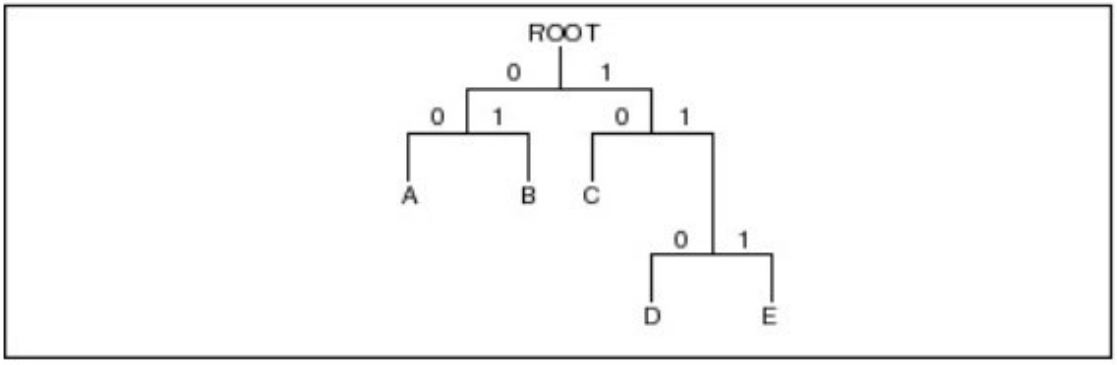

Abbildung 1: Shannon-Fano Baum [Mark Nelson: The Data Compression Book, S 30]

2.3 Huffmann-Kodierung

Bei der Huffman Kodierung wird wieder die Zeichenfolge nach Häufigkeit aufgelistet, nur in diesem Fall aufsteigend vom seltensten Zeichen aus beginnend. Der darstellbare Baum wird von den Blättern nach oben bis zur Wurzel (bottom-up) aufgebaut. Der Informationsgehalt je Zeichen kann wie folgt dargestellt werden:

$$I(z_i) = log_2 \frac{1}{p_i}$$

wobei p_i die Auftrittswahrscheinlichkeit des Zeichens z_i ist.
Die Entropie H wird als gewichteter Durchschnitt aller Informationsgehalte der Symbole S dargestellt:

$$H(S) = \sum_{i=1}^{n} p_i I(Z_i) = \sum_{i=1}^{n} p_i log_2 \frac{1}{p_i}$$

"Der Huffman-Code gilt als ein optimaler, eindeutig dekodierbarer Code". [1] Dieser Algorithmus wurde in den letzten Jahren am häufigsten eingesetzt, wird nun aber nach und nach durch die arithmetische Kodierung ersetzt, da der Huffman-Code bei ungleichen Wahrscheinlichkeiten nicht sehr effizient komprimiert

2.4 arithmetische Kodierung

Die arithmetische Kodierung berechnet bereits beim 1. Lesen - on the fly - ohne dass die Codes der Zeichenfolgen bekannt sind. Die Daten werden als Intervall der rationalen Zahlen dargestellt. Ausgehend vom Intervall [0,1) wird es bei jedem weiteren Symbol verkleinert. Problematisch bei arithmetischer Kodierung ist das Ende des Kodierungsintervalls, dh der Punkt an dem alle Zeichen codiert sind, wobei sich dies zB mit dem Befehl eof(end-of-file) lösen lässt.

[1] Herold, Helmut; Lurz, Bruno; Wohlrab, Jürgen: Grundlagen der Informatik. 2. aktualisierte Auflage. München: Pearson 2012, Seite 756

2.5 Lempel-Ziv-Kodierung

Lempel-Ziv Kodierung ist ein dynamisches Verfahren, da das Wörterbuch sowohl bei der Codierung als auch bei der Decodierung implizit aus den zu verschlüsselnden Daten generiert wird. Ein Vorteil ist,dass bei der Decodierung keine zusätzliche Information notwendig ist, also nicht wie bei Huffman und arithmetischer Kodierung eine Kodierungstabelle mitübertragen wird. Datenkompression wird bei allen Lempel-Ziv-Kodierungen dadurch erlangt, dass längere Zeichenketten durch kürzere Codes ersetzt werden.

2.5.1 LZ77 & LZSS

Komprimieren von Wiederholungen (implicit dictionaries) - dh dass immer wieder Zeichengruppen auf bereits vorhandene Codes überprüft werden und lediglich Zeiger gesetzt werden, die die bereits verarbeitete Datenfolge repräsentieren.

2.5.2 LZ78, LZC & LZW

Erzeugt ein Wörterbuch aus Teilfolgen, dh direkt beim Komprimieren werden die Wörterbücher aus den Teilfolgen generiert. In diesen Teilfolgen treten die zu komprimierenden Zeichenfolgen auf. wenn eine Zeichenfolge bereits vorhanden ist, wird sie durch den Index markiert und im Eintrag ersetzt, dieses Wörteruch wird dynamisch generiert.

3 Schlussfolgerung

Jede Art der Komprimierung bringt ihre eigenen Vor- und Nachteile mit sich. Je nach Anwendungsfeld und geforderter Nutzbarkeit gibt es einen "best-practice Algorithmus", der die meisten Vorteile bringt. Genauso wie in den meisten Forschungsbereichen gibt es noch Potential für weitere und möglicherweise effizientere Alverzeichnis

1. Herold, Helmut; Lurz, Bruno; Wohlrab, Jürgen: Grundlagen der Informatik. 2. aktualisierte Auflage. München: Pearson 2012

2. Nelson, Mark: The Data Compression Book. Redwood City: M&T Publishing, Prentice Hall International 1993