SISTEMA ADAPTATIVO PARA COMPRESSÃO DE DADOS

NEWTON FALLER

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA (M. Sc.)

Aprovada por:

Prof. Ysmar Vianna e Silva Filho
Presidente

Prof. Ivan da Costa Marques

Prof. Nelson Maculan Filho

Prof. Sergio Teixeira

RIO DE JANEIRO
ESTADO DA GUANABARA - BRASIL
DEZEMBRO DE 1973

DEDICATÓRIA

A MEUS PAIS

ADA E KURT

AGRADECIMENTO

Aos Professores JOHN HEINBOLD e CARLOS HARTMANN por algumas idéias iniciais.

Ao Professor YSMAR VIANNA pela orientação durante todo desenvolvimento do trabalho.

Ao Professor JOÃO LIZARDO pelas sugestões relativas ao teste do sistema.

A MARIA ALICE F. C. MELLO pela extrema dedicação e paciência na preparação datilografada dos originais.

RESUMO

Uma propriedade interessante é provada para árvore de Huffman.
Quaisquer dois elementos de pesos a_i e b_i filhos de um mesmo pai tem a seguinte propriedade :

se $b_i \geqslant a_i$ então

$W_m \leqslant a_i$ ou $W_m \geqslant b_i$

onde W_m é o peso de qualquer nó da árvore.

Baseado nesta propriedade, um algoritmo foi desenvolvido para atualizar dinâmicamente uma árvore de Huffman, à medida que os pesos dos seus nós terminais variam.

Utilizando-se este algoritmo, um modelo de um sistema adaptativo para compressão de dados foi implementado.

Simulações efetuadas com diversos tipos de dados levaram a resultados interessantes.

ABSTRACT

An interesting property is proven for Huffman's tree. Any two elements with weights a_i and b_i sons of a same father have the following property :

If $b_i \geqslant a_i$ then

$W_m \leqslant a_i$ or $W_m \geqslant b_i$

where W_m is the weight of any node of the tree.

Based on this property, an algorithm is developed to dynamically update Huffman's tree as weights of terminal nodes change.

Using this algorithm, a model of an adaptive system for data compression is developed.

Simulation using many types of data led to interesting results.

ÍNDICE

1. INTRODUÇÃO

Uma técnica que se tem procurado sempre aprimorar é a compressão de dados. Comprimir dados significa reduzir a redundância inerente aos conjuntos de dados a serem codificados, tendo como objetivo representar apenas a informação real contida em tal conjunto.

Nos sistemas digitais a informação é codificada através de uma seqüência de digitos binários (bits). Isto se deve ao fato de que na tecnologia atual é mais fácil reconhecer apenas dois estados (0 e 1) em um dispositivo do sistema.

Dos estudos de Teoria da Informação [1] sabe-se que a entropia de uma fonte binária ergódica pode ser definida como

$$H = - \sum_{n=1}^{n} p_i \log_2 p_i \quad \text{bits/símbolo}$$

onde n é o número de símbolos a ser codificado e p_i é a probabilidade de ocorrência do i-ésimo símbolo do conjunto de dados.

Informalmente, a entropia de uma fonte define o número mínimo de bits por símbolo necessários para codificar mensagens geradas por esta fonte. Este número, entretanto, é um limite inferior e em geral não é inteiro. Como o número de bits que representa determinado símbolo tem que ser um número inteiro, sòmente utilizando-se códigos de tamanho variável é possível aproximar-se deste limite inferior. Um método elegante para definir estes códigos, sabendo-se de antemão a probabilidade de ocorrência de cada um dos símbolos, foi dado por Huffman [2] .

Um código binário pode ser representado por uma árvore binária em que os caminhos à esquerda de cada nó correspondem aos zeros enquanto os à direita correspondem aos uns. A informação é associada aos nós terminais. Achar o símbolo correspondente a determinada codificação, significa partir da raiz da árvore seguindo à direita ou à esquerda, conforme os bits 1 ou 0 da codificação até alcançar-se um nó terminal. Existe portanto um caminho único que une a raiz da árvore a um nó terminal.

Neste trabalho considerou-se uma árvore de Huffman na qual os pesos as

sociados aos diversos nós terminais variavam com o tempo. Um algoritmo para atualizar a árvore de forma a mantê-la sempre numa árvore de Huffman, foi desenvolvido. Baseado neste algoritmo um modelo de um sistema auto-adaptativo foi implementado.

Usando este sistema, não há necessidade de se saber a priori a frequência de ocorrência de cada símbolo do conjunto. À medida que um símbolo é codificado, o sistema computa a sua frequência relativa e adapta-se de tal maneira a manter um código com mínima redundância.

2. DEFINIÇÕES

A não ser nos pontos onde for explicitamente declarado, todas as definições e notações referentes a árvores são as mesmas usadas pelo Knuth [3] .

Árvore binária (chamada árvore binária extendida pelo Knuth) considera-se um nó chamado raiz com as suas duas árvores binárias distintas.

Floresta é um conjunto de árvores.

Nó interno, incluindo a raiz, tem dois filhos, o direito e o esquerdo, que são as raizes das sub-árvores direita e esquerda, correspondentes ao nó pai.

Nós terminais, nós externos ou folhas são nós que não possuem filhos.

Ao longo deste trabalho supõe-se sempre n nós terminais. O termo nó pode designar indistintamente um nó interno ou externo. Existe um único caminho entre a raiz e cada um dos nós (interno ou externo).

Comprimento do caminho de um nó é o comprimento do caminho (número de arcos) percorrido entre a raiz e este nó.

Comprimento do caminho de uma árvore é a soma dos comprimentos do caminho de todos os nós terminais. Assim, se l_j é o comprimento do caminho do j-ésimo nó terminal, então $\sum l_j$ é o comprimento do caminho da árvore.

Um nó está no nível k quando o comprimento do caminho deste nó é igual

a k. A raiz está no nível zero e é considerado o mais alto nível.

Em muitos problemas associa-se a cada nó terminal um peso w. Comprimento ponderado do caminho de uma árvore ou simplesmente custo de uma árvore define-se como $\sum w_j l_j$ onde w_j e l_j são respectivamente o peso e o comprimento do caminho do j-ésimo nó terminal. Comprimento ponderado médio do caminho de uma árvore ou custo normalizado de uma árvore é dado por $\sum w_j l_j / \sum w_j$.

A árvore de menor custo possível de se construir com os mesmos n nós terminais chama-se árvore ótima.

Lema 2.1. : O número total de nós internos numa árvore binária é um menos que o número total de nós terminais [3] . Associemos também a cada nó interno um peso que seja a soma dos pesos de seus nós filhos.

A árvore construída com estes pesos associados a cada um de seus nós é a árvore que será considerada deste ponto em diante. Os pesos dos nós internos serão designados por w'.

Lema 2.2. : A soma dos n-1 pesos dos n-1 nós internos é o comprimento ponderado da árvore ou o custo da árvore [3] .

Assim $\sum w'_i = \sum w_j l_j$.

3. A ÁRVORE DE HUFFMAN

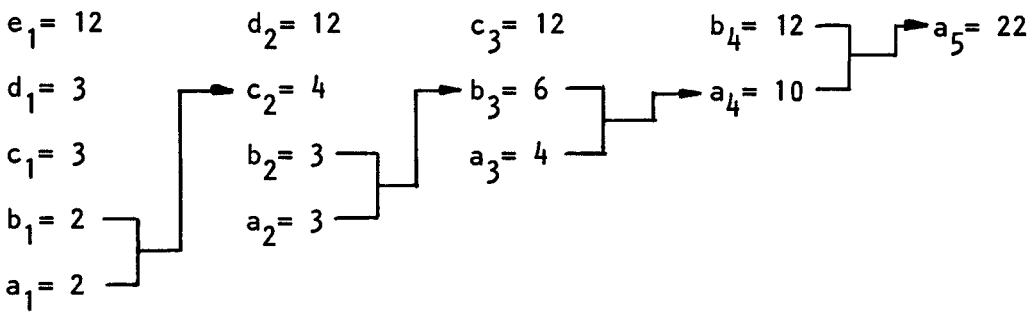
A árvore de Huffman é aquela construida através do algoritmo de Huffman 2 . O algoritmo de Huffman é o seguinte :

1) Colocar os n pesos dos nós terminais em ordem ascendente ;

2) Tirar os dois de menor valor da sequência, somá-los e introduzir esta soma novamente na sequência de forma a manter a ordem ascendente. Associar os dois valores retirados da sequência a nós filhos de uma sub-árvore cuja raiz tem peso igual à soma destes valores.

3) Repetir o passo 2 até que exista apenas um valor na sequência. Este valor é então associado à raiz da árvore.

(a) ALGORITMO DE HUFFMAN



(b) ÁRVORE DE HUFFMAN

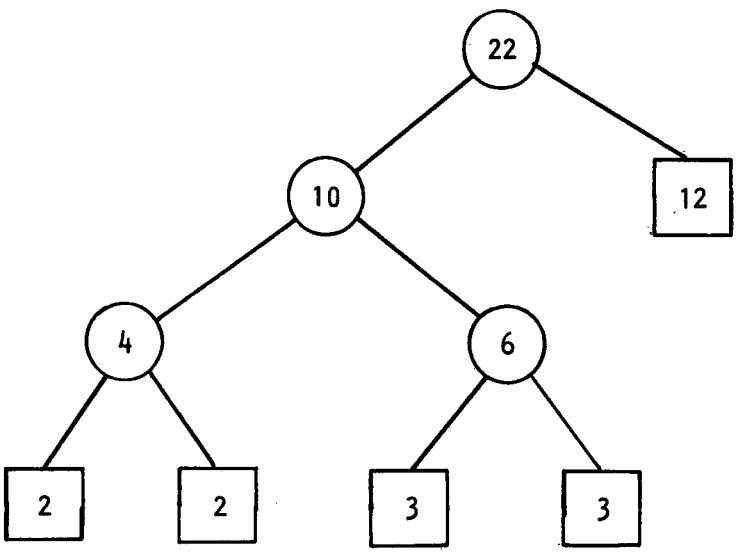


FIGURA 1. EXEMPLO DO ALGORITMO DE HUFFMAN E CONSTRUÇÃO DA ÁRVORE CORRESPONDENTE.