An Improved Bound for Weight-Balanced Tree

YASUICHI HORIBE

Department of Information Science, Faculty of Engineering,
Shizuoka University, Johoku 3-5-1, Hamamatsu, Japan

An improved upper bound is obtained on the averaged path length of an alphabetical binary tree (or equivalently on the averaged word length of the alphabetical binary code) which is constructed by weight balancing.

The object of this paper is to improve the results due to Rissanen [5] according to his interesting line of argument.

1. BINARY TREE

In a binary (rooted and ordered) tree, (Knuth [4]), suppose that there are n external nodes $1,..., n$, an external node (leaf) being one without son. We have, therefore, $n - 1$ internal nodes, an internal node having just two sons (see Fig. 1). The (alphabetical) binary tree can be viewed as a graphic representation of successive dichotomies of the set $\{1,..., n\}$, each resulting subset consisting of *consecutive* integers. Thus an internal node corresponds to a subset $\{i, i + 1,..., j\}$ such that $i < j$, and the subtree whose root is this internal node has external nodes $i, i + 1,..., j$. The internal node corresponding to $\{i, i + 1,..., j\}$ may therefore be denoted by (i, j), $(i, i) = i =$ external node i.

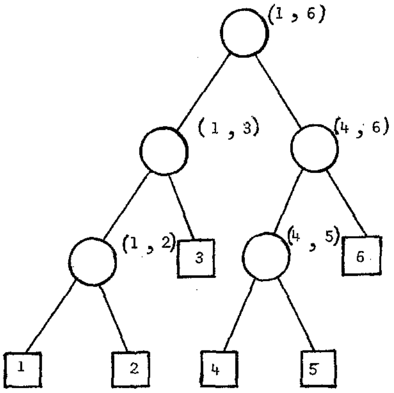

FIG. 1. Weight balanced tree for $p_1 = p_3 = p_4 = p_6 = \epsilon$, $p_2 = p_5 = \frac{1}{2} - 2\epsilon$.

ISSN 0019-9958

Each external node in the tree can be naturally identified with a binary word ($0 =$ left branching, $1 =$ right branching). The length l_i of the word for external node i is the length of the unique path from i to the root of the tree.

Suppose further that external node i has probability or weight $p_i > 0$, $i = 1,..., n$, $p_1 + \cdots + p_n = 1$. The averaged word length or the averaged path length is defined by $L = \sum_{i=1}^{n} p_i l_i$. Let us denote by I the set of all internal nodes, and write $p(i, j) = p_i + p_{i+1} + \cdots + p_j$, $i \leqslant j$. Then it is easily seen that

$$L = \sum_{(i,j)\in I} p(i, j). \tag{1}$$

2. Entropy

The following is the well-known decomposition of entropy $H = H(p_1 ,..., p_n) = -\sum_{i=1}^{n} p_i \log p_i$ [1].

$$\begin{aligned} H(p_1 ,..., p_n) &= H(p(1, k), p(k + 1, n)) \\ &\quad + p(1, k)\, H\left(\frac{p_1}{p(1, k)} ,..., \frac{p_k}{p(1, k)}\right) \\ &\quad + p(k + 1, n)\, H\left(\frac{p_{k+1}}{p(k + 1, n)} ,..., \frac{p_n}{p(k + 1, n)}\right), \end{aligned}$$

where $1 \leqslant k < n$ and the log base is 2. The successive applications of this decomposition yield the following weighted sum of binary entropies:

$$H = \sum_{(i,j)\in I} p(i, j)\, H\left(\frac{p(i, k)}{p(i, j)}, \frac{p(k + 1, j)}{p(i, j)}\right), \qquad i \leqslant k < j, \tag{2}$$

where (i, k), $(k + 1, j)$ are left son and right son, respectively (external or internal), of the internal node (i, j).

By (1) and (2) we have

$$L - H = \sum_{(i,j)\in I} p(i, j) \left\{1 - H\left(\frac{p(i, k)}{p(i, j)}, \frac{p(k + 1, j)}{p(i, j)}\right)\right\}.$$

We can prove that the inequality $0 \leqslant 1 - H(p, q) \leqslant | p - q |^{\nu}, p + q = 1$, holds for $\nu \leqslant 2$, but the case $\nu = 1$ is convenient for our purposes, which gives

$$\begin{aligned} &H \leqslant L \leqslant \Delta + H, \\ &\Delta = \sum_{(i,j)\in I} | p(i, k) - p(k + 1, j)|, \qquad i \leqslant k < j. \end{aligned} \tag{3}$$

3. Weight Balancing

Gilbert and Moore [2] have shown that in the optimum alphabetical binary tree (the one with minimum L), we have $L < 2 + H$. Hu and Tucker [3] give a remarkable algorithm for constructing an optimum alphabetical binary tree.

In some cases, however, especially when n is large as in the searching problem [4], a simpler and more straight-forward algorithm might be desirable if a nearly optimum tree could be constructed. Equation (3) suggests such an algorithm. For the root $(1, n) \in I$, take $(1, k)$, $(k + 1, n)$ as its left and right sons, respectively, such that $| p(1, k) - p(k + 1, n)| = \min_{1 \leqslant l < n} | p(1, l) - p(l + 1, n)|$; i.e., k is chosen in order that left and right weights are most balancing.

In general for $(i, j) \in I$, take (i, k), $(k + 1, j)$ as its left and right sons, respectively, such that

$$| p(i, k) - p(k + 1, j)| = \min_{i \leqslant l < j} | p(i, l) - p(l + 1, j)|. \tag{4}$$

This "top-down" algorithm for constructing a binary tree will be called weight balancing. We denote by $\Delta(i, j)$ the minimum value of (4).

4. Upper Bound

Theorem. *The binary tree constructed by weight balancing gives $\Delta \leqslant \delta$ (hence $L \leqslant \delta + H$).*

$$\delta = \sum_{k=1}^{n-1} \max\{p_k , p_{k+1}\} - p_{\min} , \; p_{\min} \equiv \min_i p_i .$$

Therefore $\delta \leqslant 2 - (n + 2) p_{\min}$.

Proof. For the internal node (i, j), $i < j$, in the tree constructed by weight balancing, suppose $\Delta(i, j) = | p(i, k) - p(k + 1, j)|$ for some k, $i \leqslant k < j$. First consider what happens when $p(i, k) \geqslant p(k + 1, j)$. If $i < k < j$ ("left-interior"), then we must have $p(i, k - 1) < p(k, j)$, otherwise it would violate the weight balancing rule. Hence $\Delta(i, j) \leqslant p(k, j) - p(i, k - 1)$ by the minimality of $\Delta(i, j)$. We see

$$\begin{aligned} \Delta(i, j) &\leqslant p(k, j) - p(i, k - 1) \\ &= (p_k + p(k + 1, j)) - (p(i, k) - p_k) \\ &= 2p_k - \Delta(i, j), \end{aligned}$$

hence $\Delta(i, j) \leqslant p_k$ for this left-interior case. If $k = i$ ("left-boundary"), clearly $\Delta(i, j) = p_k - p(k + 1, j) \leqslant p_k - p_{\min} < p_k$.

When $p(i, k) < p(k+1, j)$, by the left-right symmetry we have $\Delta(i, j) \leqslant p_{k+1}$ if $i < k+1 < j$ ("right-interior") and $\Delta(i, j) \leqslant p_{k+1} - p_{\min} < p_{k+1}$ if $k+1 = j$ ("right-boundary"). Upper bounding each term $\Delta(i, j)$ of Δ by p_k or p_{k+1}, we have, since $|I| = n - 1$,

$$\Delta \leqslant \sum_{k=1}^{n-1} \max\{p_k, p_{k+1}\}.$$

Since in any tree there exists at least one internal node of the form $(i, i+1)$, and hence at least one boundary case occurs in the construction of the tree, we have $\Delta \leqslant \delta$ by subtracting $p_{\min}$, completing the proof.

As a corollary we have $L \leqslant H + 1 - 2p_{\min}$ for monotone $p_1, \ldots, p_n$.

The proof above suggests the possibility of obtaining a still sharper upper bound, but the bound so obtained seems to have a complex form.

It appears that only the simplest cases give the equality $\Delta = \delta$. For example, take $n = 2$, or take $n = 3$, $p_1 = p$, $p_2 = 1 - 2p$, $p_3 = p$ $(0 < p \leqslant \frac{1}{3})$, the latter giving $\Delta = \delta = 2 - 5p$. Observe that the example in Fig. 1 is easily seen to have $\Delta \approx 2$, $L \approx \delta + H$, if ϵ is small.

Acknowledgment

The author thanks N. Michibata for several discussions on search trees.

Received: June 18, 1976

References

1. R. B. Ash, "Information Theory," Interscience, New York, 1965.
2. E. N. Gilbert and E. F. Moore, Variable-length binary encodings, *Bell System Tech. J.* 38 (1959), 933–967.
3. T. C. Hu and A. C. Tucker, Optimal computer search trees and variable-length alphabetical codes, *SIAM J. Appl. Math.* 21 (1971), 514–532.
4. D. E. Knuth, "The Art of Computer Programming," Vol. 3, Sorting and Searching, Addison-Wesley, Reading, Mass., 1973.
5. J. Rissanen, Bounds for weight balanced trees, *IBM J. Res. Develop.* **17** (1973), 101–105.