2004－AL－94 (15)
2004／3／19

ハフマン木の最適領域の隣接性について

大西 建輔

電気通信大学 大学院情報システム学研究科,
〒 182-8585 調布市調布ケ丘 1-5-1,
E-mail: onishi@hol.is.uec.ac.jp

あらまし: ハフマン木は有用なデータ構造としてこれまで多くの研究がなされている. キーとその重みが与えられた場合にハフマン木は簡単に計算をすることができることはよく知られている. 重み付き拡張二分木が最適であるとは, 他の全ての拡張二分木の中でその二分木が最も重み付き路長和を最小にする場合である. 拡張二分木の最適領域とは, 重み空間の領域であり, その中に含まれる重みはその二分木を最適とする.

本論文では, ハフマン木の最適領域の性質についての研究を行う. レベルベクタとは, 拡張二分木のそれぞれの葉までの路長をベクトルとして表現した形式であり, 本稿ではレベルベクタとハフマン木を同一視する. まず, どの最適領域も非空かつ、凸集合であることを示す. また, 2 つのハフマン木の最適領域が隣接しているとは, ある重みが存在し, その重みでその 2 つのハフマン木だけが最適になると定義する. この最適性の必要十分条件が次の 2 つであることを証明する: 1) 葉のレベル差が高々1 である; 2) その 2 つのハフマン木のレベルベクタの和が他のどの 2 つのレベルベクタの和としても表されない.

Adjacency of optimal region of Huffman tree

Kensuke Onishi

Graduate School of Information Systems, University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan
E-mail: onishi@hol.is.uec.ac.jp

Abstract: Huffman tree has been studied as useful data structure. For given keys and their weights, the tree structure can be constructed easily. A extended binary tree with the weights is *optimal* if the tree has smallest weighted external path length among all extended binary trees. Consider the weight space and an extended binary tree. *Optimal region* of the extended binary tree is defined by a region of the weight space such that the tree with any weight in the region is optimal.

In this paper we investigate properties of optimal regions. Huffman tree is expressed as *level vector* which is defined by integer vector whose element is the path length to each node of the tree. We show that each optimal region is non-empty and convex through level vector. Two Huffman trees are *adjacent* if there is a weight such that the only two trees are optimal for the weight. We show that the necessary and sufficient condition of adjacency is that 1) the difference of level between each leaf is at most one; 2) the sum of two level vectors of trees is not equal to the sum of any other two level vectors.

1 Introduction

Huffman tree is a fundamental and popular data structure. The tree has many applications [2, pp. 402-406]. All leaves of Huffman tree are labelled. In this paper Huffman tree is regarded as a positive integer vector $\boldsymbol{l} = (l_1, \cdots, l_n)$, called *level vector*. There is an extended binary tree, denoted by $T(\boldsymbol{l})$, corresponding to this vector if $\sum_i 2^{-l_i} = 1$. Each value of l_i indicates that the leaf indexed by i has level l_i in the binary tree. In some case there are several different binary trees for $\boldsymbol{l}$. In this paper these trees are regarded as the same. The set $\mathcal{L}_n$ of all n-dimensional level vector is defined by

$$\mathcal{L}_n = \left\{ \boldsymbol{l} \in \mathbb{Z}^n_{++} \;\middle|\; \sum_{i=1}^{n} 2^{-l_i} = 1 \right\},$$

where $\mathbb{Z}^n_{++} := \{(x_1, \ldots, x_d) \,|\, x_i \in \mathbb{Z}, x_i > 0\}$. For a given weight $\boldsymbol{w} > \boldsymbol{0}$, a level vector $\boldsymbol{l}$ is *optimal* if $\boldsymbol{l} \cdot \boldsymbol{w} \leq \boldsymbol{l}' \cdot \boldsymbol{w}$ for any other $\boldsymbol{l}' \in \mathcal{L}_n$. This optimality is the same as the optimality of corresponding tree, which is a Huffman tree with $\boldsymbol{w}$. For each $\boldsymbol{l}$ *optimal* *region* $R(\boldsymbol{l})$ is defined by:

$$R(\boldsymbol{l}) = \{\boldsymbol{w} \mid \boldsymbol{l} \cdot \boldsymbol{w} \leq \boldsymbol{l}' \cdot \boldsymbol{w},\, \forall \boldsymbol{l}' \in \mathcal{L}_n\}\,.$$

Let $\boldsymbol{W}_n := \{\boldsymbol{w} = (w_1, \ldots, w_n) \mid w_i \in \mathbb{R}, w_i > 0\}$ be the set of all weights, called *weight space.* All optimal regions for every level vectors is a subdivision of $\boldsymbol{W_n}$ (see Figure 1(left)).

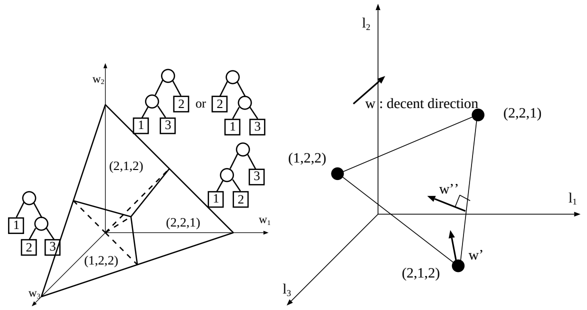


Figure 1: Optimal regions for Huffman trees with 3 leaves(left) and convex hull for $\mathcal{L}_3$(right)

We investigate properties of these regions. One problem is which $\boldsymbol{l}$s can be optimized simultaneously. Two $\boldsymbol{l}$ and $\boldsymbol{l}'$ are said to be *similar* if there exists a weight $\boldsymbol{w}$ such that $\boldsymbol{l} \cdot \boldsymbol{w} = \boldsymbol{l}' \cdot \boldsymbol{w} \leq \boldsymbol{l}'' \cdot \boldsymbol{w}, \forall \boldsymbol{l}'' \in \mathcal{L}_n$. Other problem is which $\boldsymbol{l}$s share their facet of $R(\boldsymbol{l})$. Two $\boldsymbol{l}$ and $\boldsymbol{l}'$ are said to be *adjacent* if there exists a weight $\boldsymbol{w}$ such that $\boldsymbol{l} \cdot \boldsymbol{w} = \boldsymbol{l}' \cdot \boldsymbol{w} < \boldsymbol{l}'' \cdot \boldsymbol{w}, \forall \boldsymbol{l}'' \in \mathcal{L}_n$.

This problem is regarded as optimization. Let $\boldsymbol{L}_n := \{(x_1, \cdots, x_n) \mid x_i \in \mathbb{R}\}$ be the space containing all n dimensional level vectors. In $\boldsymbol{L}_n$ a level vector is regarded as a point and a weight is as a decent direction. So, the computation of Huffman tree under a weight $\boldsymbol{w}$ is equivalent to the following optimization problem in $\boldsymbol{L}_n$:

$$\min_{\boldsymbol{l} \in \mathcal{L}_n} \boldsymbol{w} \cdot \boldsymbol{l}.$$

In Figure 1(right) when the direction is $\boldsymbol{w}'$, the level vector $(2, 1, 2)$ is optimal.

The adjacency of optimal regions is easily understood. That two optimal regions $R(\boldsymbol{l})$ and $R(\boldsymbol{l})'$ are adjacent is equivalent to that two level vectors $\boldsymbol{l}$ and $\boldsymbol{l}'$ are connected by an edge on the convex hull. In Figure 1, optimal regions $R((2, 1, 2))$ and $R((2, 2, 1))$ share a facet, while level vectors $(2, 1, 2)$ and $(2, 2, 1)$ are connected by an edge of convex hull of $\mathcal{L}_3$.

In Section 2, we describe some well-known results about Huffman tree. Section 3 is main section and some properties of optimal region are shown: non-empty, convexity and *adjacency.* In Section 4, some applications of these results are described.

2 Preliminaries

In this section we state some results about Huffman tree.

We state conditions for a given tree T under weight $\boldsymbol{w}$ to be optimal or not. In [4, p.25] the following theorem is given.

> **Lemma A** A weighted extended binary tree on a sequence of weights $w_1, w_2, \ldots, w_n$ is optimal only if the following conditions are settled for any level l:
> 1) the weight of any node at a level l is greater than or equal to the weight of any node at level $l + 1$;
> 2) for any three nodes a, b, c in level l, $w(c) \leq w(a) + w(b)$.

So, we can check the optimality by computing maximum and two minimums weights in every levels.

3 Properties of optimal region

In this section we describe properties of optimal region. The optimal region of Huffman tree is non-empty, convex. We show necessary and sufficient condition of *adjacency.*

3.1 non-empty and convex

Firstly, we show that any optimal region is non-empty. Let $\boldsymbol{w}(\boldsymbol{l})$ be the positive weight for $\boldsymbol{l}$ such that its ith element is $w_i = 2^{-l_i}$ where l_i is the path length from the root node to the ith node in $T(\boldsymbol{l})$. We show that $\boldsymbol{l}$ with $\boldsymbol{w}(\boldsymbol{l})$ is optimal.

Consider an extended binary tree $T(\boldsymbol{l})$. It is shown that the $T(\boldsymbol{l})$ with $\boldsymbol{w}$ satisfies the condition of optimality in **Lemma A**. In the highest level L there are only leaf nodes and their weights are equal to 2^{-L}. In the next level $L - 1$, any internal node exists and its weight is sum of the weight of the child nodes. So, the weight of internal node is equal to $2^{-L} + 2^{-L} = 2^{-(L-1)}$. Leaf nodes in level $L - 1$ also have the weight $2^{-(L-1)}$. Thus every node in level $L-1$ has the weight $2^{-(L-1)}$. Since this step repeats until level 0, it is shown that every node has weight 2^{-l} in level l. So, it is evident that the tree satisfies the condition 1) and 2) in **Lemma A**. Thus each optimal region is non-empty.

Consider a weight $\boldsymbol{w}(\boldsymbol{l}) + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} := (\varepsilon_1, \cdots, \varepsilon_n)$ and $|\varepsilon_i| \ll 2^{l_i}$. It is similarly shown that the level vector $\boldsymbol{l}$ is optimal under $\boldsymbol{w}(\boldsymbol{l}) + \boldsymbol{\varepsilon}$. Thus the optimal region has full dimension.

Moreover, we state that optimal region is convex. Since the optimal region is defined by intersection of halfspaces from the definition, the region is convex cone.

Theorem 1 *The optimal region of any level vector is non-empty and convex. The region has full dimension.*

3.2 similarity

In this subsection we show that necessary and sufficient condition of similarity.

Firstly, we show that there exists a weight such that two level vectors $\boldsymbol{l}, \boldsymbol{l}'$ are optimal if $\boldsymbol{l} - \boldsymbol{l} \in \{-1, 0, 1\}^n$.

Lemma 1 *Let $\boldsymbol{l}, \boldsymbol{l}'$ be two n-dimensional level vectors with $\boldsymbol{l} - \boldsymbol{l}' \in \{-1, 0, 1\}^n$. Consider a weight $\frac{1}{2}\boldsymbol{w}(\boldsymbol{l}) + \frac{1}{2}\boldsymbol{w}(\boldsymbol{l}')$. The weight of any node in level l of $T(\boldsymbol{l})$ and $T(\boldsymbol{l}')$ is included in $\left[\frac{3}{4} \cdot 2^{-l}, \frac{3}{2} \cdot 2^{-l}\right]$.*

Proof: We use induction by level of $T(\boldsymbol{l})$. Consider the highest level L in the tree. At the level only leaves exist. Since the difference of leaf level between $T(\boldsymbol{l})$ and $T(\boldsymbol{l}')$ is at most one, the pair of levels is three types: $(L, L-1), (L, L), (L, L+1)$. So, their weights is only three cases: $(2^{-L} + 2^{-(L+1)})/2 = \frac{3}{2} \cdot 2^{-L}$, $(2^{-L} + 2^{-L})/2 = 2^{-L}$, $(2^{-L} + 2^{-(L-1)})/2 = \frac{3}{4} \cdot 2^{-L}$. At this level all weights are included in $\left[\frac{3}{4} \cdot 2^{-L}, \frac{3}{2} \cdot 2^{-L}\right]$.

Assume that all weights are included in $\left[\frac{3}{4}2^{-l}, \frac{3}{2}2^{-l}\right]$ at level l. We show that the weight of any node in the level $l-1$ is included in $\left[\frac{3}{4} \cdot 2^{-(l-1)}, \frac{3}{2} \cdot 2^{-(l-1)}\right]$. From the assumption, for any two nodes a, b in level l their weight $w(a), w(b)$ satisfies the following inequalities: $\frac{3}{4} \cdot 2^{-l} \leq w(a), w(b) \leq \frac{3}{2} \cdot 2^{-l}$. So the weight of internal node is given by the sum of the weights of two child nodes, the weight is bounded:

$$\frac{3}{4} \cdot 2^{-(l-1)} \leq w(a) + w(b) \leq \frac{3}{2} \cdot 2^{-(l-1)}.$$

Moreover, it is similarly shown that the weight of leaves in level $l-1$ is also bounded. Consequently, all nodes in level $l-1$ are included in $\left[\frac{3}{4} \cdot 2^{-(l-1)}, \frac{3}{2} \cdot 2^{-(l-1)}\right]$. □

By this lemma, we can check the condition in **Lemma A**. Since the minimal weight in level l is $\frac{3}{4} \cdot 2^{-l}$ and the maximal weight in level $l+1$ is $\frac{3}{2} \cdot 2^{-(l+1)}$, the condition 1) is settled for any level.

Second condition is also shown from the lemma. It is enough to show only the most extreme case. Since every weight at level l is included in $\left[\frac{3}{4} \cdot 2^{-l}, \frac{3}{2} \cdot 2^{-l}\right]$, the extreme case is the smallest weight $w(a) = w(b) = \frac{3}{4} \cdot 2^{-l}$ and the largest weight $w(c) = \frac{3}{2} \cdot 2^{-l}$. The inequality $w(c) \leq w(a) + w(b)$ holds in the case.

From the discussion above, we show the following lemma.

Lemma 2 *Let $\boldsymbol{l}, \boldsymbol{l}'$ be level vectors such that $\boldsymbol{l} - \boldsymbol{l}' \in \{-1, 0, 1\}^n$. Two extended binary trees $T(\boldsymbol{l})$ and $T(\boldsymbol{l}')$ with $\frac{1}{2}\boldsymbol{w}(\boldsymbol{l}) + \frac{1}{2}\boldsymbol{w}(\boldsymbol{l}')$ are optimal.*

Similarly, we can show that there exists a weight such that several level vectors with the weight is optimal.

Lemma 3 *Let $\boldsymbol{l}^k(l_1^k, \ldots, l_n^k)$ $(k = 1, \ldots, m)$ be level vectors such that $\boldsymbol{l}^k - \boldsymbol{l}^j \in \{-1, 0, 1\}^n$ for any k and j. Let $\boldsymbol{w}(\boldsymbol{l}^1, \ldots, \boldsymbol{l}^m)$ be a weight whose ith component is defined by*

$$w_i = \begin{cases} 2^{l_i^k} & (l_i^k = l_i^j \textit{ for any } k, j) \\ \frac{1}{2}\left(2^{l_i^k} + 2^{l_i^j}\right) & (\textit{otherwise}) \end{cases}.$$

Any $\boldsymbol{l}^k$ with $\boldsymbol{w}(\boldsymbol{l}^1, \ldots, \boldsymbol{l}^m)$ are optimal ($k = 1, \ldots, m$).

Proof: For each pair of $\boldsymbol{l}^k$ and $\boldsymbol{l}^j$, the difference of level of leaves is at most 1. The values of l_i^ks are only two case: 1) all l_i^k has the same value; 2) the values of l_i^ks are divided into l and $l+1$. It is shown that $T(\boldsymbol{l}^k)$s with $\boldsymbol{w}(\boldsymbol{l}^1, \ldots, \boldsymbol{l}^m)$ are optimal by similar discussion as above. □

Secondly, we show the inverse of lemma 2.

Lemma 4 *If two level vectors are similar, then the difference of level of each nodes are at most 1.*

Proof: Fix a level vector $\boldsymbol{l}$ and its Huffman tree $T(\boldsymbol{l})$, whose optimal region is denoted by $R(\boldsymbol{l})$.

Consider the conditions in **Lemma A**, which are necessary conditions for Huffman tree:

1. for any three nodes a, b, c in level l, $w(c) \leq w(a) + w(b)$;
2. for any node x in level l and y in level $l+1$, $w(x) \geq w(y)$.

We show that there exists a level vector $\boldsymbol{l}'$ such that $\boldsymbol{l} - \boldsymbol{l}' \in \{-1, 0, 1\}^n$ from each condition above.

Let $L(a)$ be the set of leaves included in the subtree with root node a. The weight of internal node a is the sum of the weights of leaves in $L(a)$.

Since $L(a)$, $L(b)$ and $L(c)$ are disjoint, the first condition becomes

$$\sum_{i \in L(c)} w_i \leq \sum_{i \in L(a)} w_i + \sum_{i \in L(b)} w_i. \qquad (1)$$

When the inequality (1) is regarded as $\boldsymbol{w} \cdot \boldsymbol{c} \leq 0$, each element of $\boldsymbol{c}$ is 0 or ± 1. Without loss of generality, nodes a and b are sibling in $T(\boldsymbol{l})$. Let T' be a binary tree which is obtained by interchanging the subtree with root node c and the subtree with root node the parent of a and b in $T(\boldsymbol{l})$ (see Figure 2). This tree T' is different from $T(\boldsymbol{l})$. Let $\boldsymbol{l}'$ be a level vector of T'. It is shown that there exists T' whose level vector satisfies $\boldsymbol{l} - \boldsymbol{l}' \in \{-1, 0, 1\}^n$.

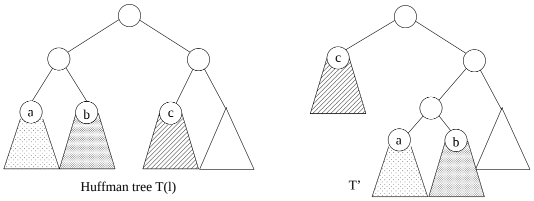


Figure 2: Huffman tree $T(\boldsymbol{l})$ and T'

The second condition also gives an inequality:

$$\sum_{i \in L(x)} w_i \geq \sum_{i \in L(y)} w_i. \qquad (2)$$

When x is the parent of y, $L(x) \supsetneqq L(y)$ is settled, otherwise $L(x) \cap L(y) = \emptyset$. Consider the case that x is parent of y. The inequality (2) becomes $\sum_{i \in L(x) \setminus L(y)} w_i \geq 0$. As every w_i is positive, this inequality gives trivial bound. Consider another case. For the inequality (2), let T'' be a binary tree which is obtained by interchanging the subtree x and the subtree y in $T(\boldsymbol{l})$. Let $\boldsymbol{l}''$ be a level vector of T'' (see Figure 3). It is shown that there exists T'' whose level vector satisfies $\boldsymbol{l} - \boldsymbol{l}' \in \{-1, 0, 1\}^n$.

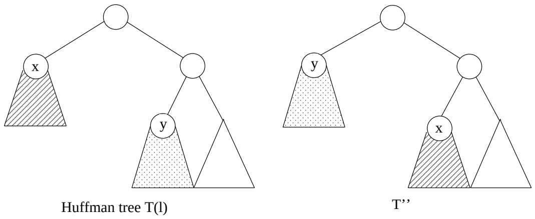


Figure 3: Huffman tree $T(\boldsymbol{l})$ and T''

Let $R'(\boldsymbol{l})$ be a polytope determined by these inequalities (1), (2). Since $R'(\boldsymbol{l})$ is defined by a part of inequalities of $R(\boldsymbol{l})$, non-empty and full dimension, $R'(\boldsymbol{l})$ contains $R(\boldsymbol{l})$. The conditions in **Lemma A** is necessary condition, the region $R'(\boldsymbol{l})$ is subset of $R(\boldsymbol{l})$. Finally, $R(\boldsymbol{l})$ is equal to $R'(\boldsymbol{l})$. The optimal region of $\boldsymbol{l}$ is defined only by these inequalities (1), (2) whose level vectors satisfy $\boldsymbol{l} - \boldsymbol{l}' \in \{-1, 0, 1\}^n$. □

[**Remark**] For one level vector $\boldsymbol{l}$ there exist several $T(\boldsymbol{l})$s. Conversely, when a Huffman tree T is given, $\boldsymbol{l}$ is uniquely determined.

Finally, the similarity can be characterized.

Theorem 2 *Two level vector $\boldsymbol{l}$ and $\boldsymbol{l}'$ are similar if and only if $\boldsymbol{l} - \boldsymbol{l}' \in \{-1, 0, 1\}^n$.*

3.3 adjacency

Two level vectors are *similar* if and only if the difference of the level of each leaves is at most 1. In this section we show necessary and sufficient conditions that two level vectors are adjacent.

Firstly, we state non-adjacent condition.

Lemma 5 *Consider level vector $\boldsymbol{l}, \boldsymbol{l}'$. If there are two level vectors $\boldsymbol{m}, \boldsymbol{m}'$ such that $\boldsymbol{l} + \boldsymbol{l}' = \boldsymbol{m} + \boldsymbol{m}'$, then $\boldsymbol{l}$ and $\boldsymbol{l}'$ are not adjacent.*

Proof: Consider any weight $\boldsymbol{w}$ such that $\boldsymbol{l} \cdot \boldsymbol{w} = \boldsymbol{l}' \cdot \boldsymbol{w}$.

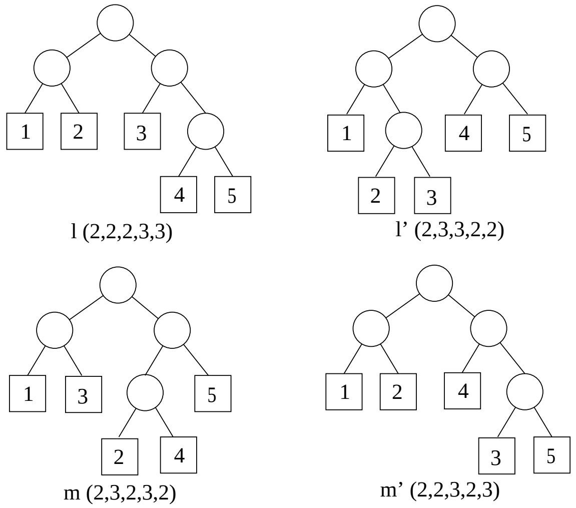
Compute inner product $\boldsymbol{l} + \boldsymbol{l}'$ and the weight $\boldsymbol{w}$:

$$\begin{aligned} (\boldsymbol{l} + \boldsymbol{l}') \cdot \boldsymbol{w} &= (\boldsymbol{m} + \boldsymbol{m}') \cdot \boldsymbol{w} \\ 2\boldsymbol{l} \cdot \boldsymbol{w} &= \boldsymbol{m} \cdot \boldsymbol{w} + \boldsymbol{m}' \cdot \boldsymbol{w} \end{aligned}$$

Since the every element of level vectors and weight $\boldsymbol{w}$ are non-negative, $\boldsymbol{l} \cdot \boldsymbol{w}$, $\boldsymbol{m} \cdot \boldsymbol{w}$ and $\boldsymbol{m}' \cdot \boldsymbol{w}$ are non-negative. Thus the following inequalities are settled:

$$\boldsymbol{l} \cdot \boldsymbol{w} \geq \boldsymbol{m} \cdot \boldsymbol{w} \text{ or } \boldsymbol{l} \cdot \boldsymbol{w} \geq \boldsymbol{m}' \cdot \boldsymbol{w}.$$

When inequality is settled, $\boldsymbol{l}$ is not optimal at $\boldsymbol{w}$. When equality is settled, $\boldsymbol{l} \cdot \boldsymbol{w} = \boldsymbol{m} \cdot \boldsymbol{w} = \boldsymbol{m}' \cdot \boldsymbol{w}$. In this case, $\boldsymbol{l}' \cdot \boldsymbol{w}$ has the same value as $\boldsymbol{l} \cdot \boldsymbol{w}$. Thus the value of inner product between four level vectors and $\boldsymbol{w}$ is the same. Since only $\boldsymbol{l}$ and $\boldsymbol{l}'$ are not only optimal, $\boldsymbol{l}$ and $\boldsymbol{l}'$ is not adjacent. □



l (2,2,2,3,3)　l' (2,3,3,2,2)

m (2,3,2,3,2)　m' (2,2,3,2,3)

Figure 4: *similar* Huffman trees

Figure 4 shows four Huffman trees. These Huffman trees are simultaneously optimal under $(1/4, 3/16, 3/16, 3/16)$. In this case the pairs ($\boldsymbol{l}$, $\boldsymbol{l}'$), ($\boldsymbol{m}$, $\boldsymbol{m}'$) are non-adjacent. Other pairs are adjacent. So, there is a hyperplane whose normal vector is $\boldsymbol{w}$ such that the hyperplane is supporting hyperplane of convex hull of $\mathcal{L}_n$.

The following lemma is the inverse of Lemma 5.

Lemma 6 *If level vector $\boldsymbol{l}, \boldsymbol{l}'$ are non-adjacent, then the either of the following holds:*

1. *$\exists i$ s.t. $|l_i - l_i'| \geq 2$;*
2. *Two level vectors $\exists \boldsymbol{m}, \boldsymbol{m}'$ such that $\boldsymbol{l} + \boldsymbol{l} = \boldsymbol{m} + \boldsymbol{m}'$;*

where $l_i (l_i')$ is the ith component of $\boldsymbol{l} (\boldsymbol{l}')$, respectively.

Proof: When the first statement does not hold, we show that second statement holds.

Since statement 1 is not settled, for any i $|l_i - l_i'| \leq 1$. From this and Theorem 2, $\boldsymbol{l}$ and $\boldsymbol{l}'$ are similar. So, there exists a weight $\boldsymbol{w}^*$ such that $\boldsymbol{l} \cdot \boldsymbol{w}^* = \boldsymbol{l}' \cdot \boldsymbol{w}^*$ and $\boldsymbol{l}$ and $\boldsymbol{l}'$ are optimal.

Since $\boldsymbol{l}$ and $\boldsymbol{l}'$ are non-adjacent, there is another level vector $\boldsymbol{m}$ such that $\boldsymbol{m} \cdot \boldsymbol{w}^* = \boldsymbol{l} \cdot \boldsymbol{w}^*$. Since $\boldsymbol{l}, \boldsymbol{l}'$ and $\boldsymbol{m}$ are simultaneously optimal under $\boldsymbol{w}^*$, the difference of each level of leaves is at most one. The indices are divided into 4 types:

$$\begin{aligned}
I_1 &= \{i \mid l_i = l_i' = m_i\}, \\
I_2 &= \{i \mid l_i = l_i', m_i = l_i \pm 1\}, \\
V_1 &= \{i \mid l_i \neq l_i', l_i = m_i\}, \\
V_2 &= \{i \mid l_i \neq l_i', l_i' = m_i\},
\end{aligned}$$

where m_i is ith coordinate of $\boldsymbol{m}$.

Firstly, we show that I_2 is empty set. Consider the optimality of $\boldsymbol{l}$, $\boldsymbol{l}'$ and $\boldsymbol{m}$ at $\boldsymbol{w}^* := \frac{1}{2}\{\boldsymbol{w}(\boldsymbol{l}) + \boldsymbol{w}(\boldsymbol{l}')\}$.

$$\begin{aligned}
&\boldsymbol{m} \cdot \boldsymbol{w}^* = \sum_{i=0}^{n} m_i \cdot \frac{1}{2}\left(2^{-l_i} + 2^{-l_i'}\right) \\
&= \sum_{i \in I_1} \frac{m_i}{2}\left(2^{-l_i} + 2^{-l_i'}\right) + \sum_{i \in I_2} \frac{m_i}{2}\left(2^{-l_i} + 2^{-l_i'}\right) \\
&\quad + \sum_{i \in V_1} \frac{m_i}{2}\left(2^{-l_i} + 2^{-l_i'}\right) + \sum_{i \in V_2} \frac{m_i}{2}\left(2^{-l_i} + 2^{-l_i'}\right) \\
&= \sum_{i \in I_1} \frac{l_i}{2}\left(2^{-l_i+1}\right) + \sum_{i \in I_2} \frac{m_i}{2}\left(2^{-l_i+1}\right) \\
&\quad + \sum_{i \in V_1} \frac{l_i}{2}\left(2^{-l_i} + 2^{-l_i'}\right) + \sum_{i \in V_2} \frac{l_i'}{2}\left(2^{-l_i} + 2^{-l_i'}\right)
\end{aligned}$$

This value is equal to $\boldsymbol{l} \cdot \boldsymbol{w}^* = \sum_{i=0}^{n} l_i/2 \left(2^{-l_i} + 2^{-l_i'}\right)$. The following equality is shown:

$$\sum_{i \in I_2} (l_i - m_i) 2^{-l_i+1} = \sum_{i \in V_2} (l_i' - l_i)\left(2^{-l_i} + 2^{-l_i'}\right). \tag{3}$$

Next, we consider a weight $\boldsymbol{w}(\boldsymbol{l}, \boldsymbol{l}', \boldsymbol{m})$ in Lemma 3. $\boldsymbol{l}$, $\boldsymbol{l}'$ and $\boldsymbol{m}$ with $\boldsymbol{w}(\boldsymbol{l}, \boldsymbol{l}', \boldsymbol{m})$ are optimal simultaneously. If i is included in I_1 or V_1 or V_2, the weight of ith coordinate is equal to w_i in the weight $\boldsymbol{w}^*$. If i is included in I_2, then $w_i = \frac{1}{2}\left(2^{-l_i} + 2^{-m_i}\right)$. From $\boldsymbol{l} \cdot \boldsymbol{w}(\boldsymbol{l}, \boldsymbol{l}', \boldsymbol{m}) = \boldsymbol{m} \cdot \boldsymbol{w}(\boldsymbol{l}, \boldsymbol{l}', \boldsymbol{m})$, the following equation is shown:

$$\sum_{i \in I_2} (l_i - m_i)\left(2^{-l_i} + 2^{-m_i}\right) = \sum_{i \in V_2} (l_i' - l_i)\left(2^{-l_i} + 2^{-l_i'}\right). \tag{4}$$

We can get the following equality from equation (3) and (4).

$$\sum_{i \in I_2} (l_i - m_i)(2^{-m_i} - 2^{-l_i}) = 0 \tag{5}$$

The I_2 are divided into two sets.

$$\begin{aligned} I_2^+ &= \{i \mid l_i - m_i = 1\}, \\ I_2^- &= \{i \mid l_i - m_i = -1\}. \end{aligned}$$

Rewrite the equation (5):

$$\begin{aligned} 0 &= \sum_{i \in I_2^+} (l_i - m_i)(2^{-m_i} - 2^{-l_i}) \\ &\qquad + \sum_{i \in I_2^-} (l_i - m_i)(2^{-m_i} - 2^{-l_i}) \\ &= \sum_{i \in I_2^+} (2^{-l_i+1} - 2^{-l_i}) + \sum_{i \in I_2^-} -(2^{-l_i-1} - 2^{-l_i}) \\ &= \sum_{i \in I_2^+} 2^{-l_i} + \sum_{i \in I_2^-} 2^{-l_i-1} \end{aligned}$$

The each term in the right hand of the equation above is positive. So, I_2^+ and I_2^- are empty to satisfy the equation. Thus, $I_2 = I_2^+ \cup I_2^-$ is also empty.

Secondly, we show that $\boldsymbol{m}' = (m_1', \ldots, m_n') := \boldsymbol{l} + \boldsymbol{l}' - \boldsymbol{m}$ is a level vector of Huffman tree, i.e. $\boldsymbol{m}' \cdot \boldsymbol{w} = \boldsymbol{l} \cdot \boldsymbol{w}$ and $\sum_i 2^{-m_i'} = 1$.

That the value of $\boldsymbol{m}' \cdot \boldsymbol{w}$ is equal to $\boldsymbol{l} \cdot \boldsymbol{w}$ is stated from the definition of $\boldsymbol{m}'$.

$$\begin{aligned} \boldsymbol{m}' \cdot \boldsymbol{w} &= \{\boldsymbol{l} + \boldsymbol{l}' - \boldsymbol{m}\} \cdot \boldsymbol{w} \\ &= \boldsymbol{l} \cdot \boldsymbol{w} + \boldsymbol{l}' \cdot \boldsymbol{w} - \boldsymbol{m} \cdot \boldsymbol{w} \\ &= \boldsymbol{l} \cdot \boldsymbol{w}. \end{aligned}$$

Since $\boldsymbol{l}, \boldsymbol{l}'$ and $\boldsymbol{m}$ correspond to Huffman trees, for each trees the sum of 2^{-l_i} is equal to 1. Let $S(I, \boldsymbol{l})$ be the sum of 2^{-l_i} where $i \in I$ and l_i is ith coordinate of $\boldsymbol{l}$. The equalities below are settled:

$$\begin{aligned} 1 &= S(I_1, \boldsymbol{l}) + S(V_1, \boldsymbol{l}) + S(V_2, \boldsymbol{l}) \\ 1 &= S(I_1, \boldsymbol{l}') + S(V_1, \boldsymbol{l}') + S(V_2, \boldsymbol{l}') \\ &= S(I_1, \boldsymbol{l}) + S(V_1, \boldsymbol{l}') + S(V_2, \boldsymbol{l}') \\ 1 &= S(I_1, \boldsymbol{m}) + S(V_1, \boldsymbol{m}) + S(V_2, \boldsymbol{m}) \\ &= S(I_1, \boldsymbol{l}) + S(V_1, \boldsymbol{l}) + S(V_2, \boldsymbol{l}') \end{aligned}$$

From these equalities we get the following relations:

$$S(V_1, \boldsymbol{l}) = S(V_1, \boldsymbol{l}'), \quad S(V_2, \boldsymbol{l}) = S(V_2, \boldsymbol{l}').$$

If $i \in V_1$, then $m_i' = l_i + l_i' - m_i = l_i'$. So, an equation $S(V_1, \boldsymbol{m}') = S(V_1, \boldsymbol{l}')$ holds. Similarly, $S(V_2, \boldsymbol{m}') = S(V_2, \boldsymbol{l})$ also holds. Thus, we can show the following equation:

$$\begin{aligned} \sum_i 2^{-m_i'} &= S(I_1, \boldsymbol{m}') + S(V_1, \boldsymbol{m}') + S(V_2, \boldsymbol{m}') \\ &= S(I_1, \boldsymbol{l}) + S(V_1, \boldsymbol{l}') + S(V_2, \boldsymbol{l}) \\ &= S(I_1, \boldsymbol{l}) + S(V_1, \boldsymbol{l}) + S(V_2, \boldsymbol{l}) = 1 \end{aligned}$$

□

Finally we show the following theorem from Lemma 5 and 6.

Theorem 3 *Two n-dimensional level vector $\boldsymbol{l}$ and $\boldsymbol{l}'$ are adjacent if and only if*

- *$\boldsymbol{l} - \boldsymbol{l}' \in \{-1, 0, 1\}^n$;*
- *for any other level vectors $\boldsymbol{m}$ and $\boldsymbol{m}'$ $\boldsymbol{l} + \boldsymbol{l}' \neq \boldsymbol{m} + \boldsymbol{m}'$.*

4 Application

In this section we describe some applications of this result.

4.1 computation of similar Huffman trees

In this section we show the method of generation of similar Huffman trees for a given Huffman tree (level vector $(l_1, \cdots, l_n)$).

The conditions of Huffman tree are given in Lemma A.

- $w(x) \leq w(y)$ for any node x in level $l+1$ and y in l;
- $w(a) + w(b) \geq w(c)$ for any nodes a, b and c in the same level.

From these conditions similar binary trees are computed. When select a pair of nodes or a triple of nodes in the condition, we can get a similar binary tree.

From the first condition the following inequality is shown:

$$\sum_{i \in L(x)} w_i \leq \sum_{j \in L(y)} w_j.$$

A similar level vector is computed by subtracting one from the value l_i $(i \in L(x))$ and adding one to the value $l_j (j \in L(y))$.

From the second condition similar binary trees can be computed similarly. In this case the inequality becomes

$$\sum_{i \in L(a)} w_i + \sum_{j \in L(b)} w_j \geq \sum_{k \in L(c)} w_k$$

for a, b and c in the same level. For the given level vector we change some parts: $l_k + 1 (k \in L(c))$ and $l_i - 1 (i \in L(a) \cup L(b))$.

The number of similar binary trees of the given tree can be counted. Let n_l and m_l be the numbers of leaves and internal nodes in level l. The following relation holds: $m_l = \sum_{k=l+1}^{L} n_k 2^{l-k}$, where L is the maximum level of the given tree. So, from the first condition the number of similar binary tree is computed:

$$\begin{aligned}
&\sum_{l=1}^{L-1}(n_l + m_l)(n_{l+1} + m_{l+1}) \\
&= \sum_{l=1}^{L-1}\left(n_l + \sum_{k=l+1}^{L} n_k 2^{l-k}\right)\left(n_{l+1} + \sum_{k=l+2}^{L} n_k 2^{l-k}\right) \\
&\sim O(Ln^2).
\end{aligned}$$

From the second condition the number is similarly counted.

$$\sum_{i=3}^{L} n_l \binom{n_l - 1}{2} \sim O(Ln^3).$$

Finally, the total number of similar binary trees is equal to the sum of numbers above:

$$O(Ln^3).$$

4.2 enumeration by reverse search

Reverse search [1] is a method of enumeration of all nodes of graph by a proper parent relation. In this section we show that all binary tree can be enumerated by reverse search.

Consider graph $G = (V, E)$ s.t. V is a set $\mathcal{L}_n$of binary tree with n leaves and there is a edge between binary trees if the trees are similar. Lexicographic order is induced in $\mathcal{L}_n$ such that for level vector $\boldsymbol{l}, \boldsymbol{l}'$ we defined order as follows:

$$\boldsymbol{l} < \boldsymbol{l}' \Leftrightarrow l_i = l'_i (i = 1, \dots, K), l_{K+1} < l'_{K+1}.$$

For each nodes in the graph, parent relation is defined as follows: parent of given vector $\boldsymbol{l}$ is the smallest level vector among all similar level vector of $\boldsymbol{l}$, denoted by parent($\boldsymbol{l}$). This relation is realized by decreasing as many prefix $l_i (i = 1, \dots, j)$ as possible i.e. check the inequality with w_1, w_2 and so on (candidates are computed by the method in previous subsection).

Enumeration is performed as follows: consider binary trees with n leaves. There is a binary tree $(1, 2, \cdots, n-1, n-1)$, which is the smallest level vector in $\mathcal{L}_n$, called *start node*. All binary tree with n leaves enumerates from the start node by parent relation.

Suppose a binary tree $\boldsymbol{l}$ is arrived in the enumeration. There is two cases: first arrival and non-first arrival.

```
First arrival:
Enumerate $\boldsymbol{l}$.
Select the largest level vector of $\boldsymbol{l}$ in the order above:
  Let $k = 1$;
  while $(k < n)$
    if Check the possibility of increasing $l_k$ then
      Consider the set of inequalities
        $w_k + \sum_{i \in S^+ \setminus \{k\}} w_i - \sum_{j \in S^-} w_j \geq 0,$
      where $S^+ := \{i \,|\, l'_i - l_i = +1\}$,
      $S^- := \{i \,|\, l'_i - l_i = -1\}$.
      Select one level vector from the set such that $S^+$ is included small index as many as possible.
    else
      $k := k + 1$;
    endif
  endwhile
If there is no smaller child rather than $\boldsymbol{l}$, move to parent($\boldsymbol{l}$).
```

Non-first arrival: In case of return from child node $\boldsymbol{l}'$. Let $\boldsymbol{l}$ be the parent of $\boldsymbol{l}$. Consider the difference $\boldsymbol{l}' - \boldsymbol{l}$. This difference becomes $(0, \dots, 0, 1, *, \cdots, *)$. Since enumerate larger node in the child nodes, the part of 0s is not changed. The following is performed .

1. the part of $*$s decrease.

2. change the first one to zero.

If these steps are not performed, there is no smaller child rather than $\boldsymbol{l}'$. Move to parent($\boldsymbol{l}$).
When there is no parent of current node, then the enumeration is finished.

5 Conclusion

In this paper we deal with properties of optimal region of Huffman tree. The tree is expressed by level vector in this paper. The region of level vector is non-empty, convex and has full-dimension. We characterize similarity and adjacency between two regions: two regions are similar if and only if the difference of level vectors is contained in $\{-1,0,1\}^n$. Two regions of $\boldsymbol{l}, \boldsymbol{l}'$ are adjacent if and only if two regions are similar and for any other level vectors $\boldsymbol{m}, \boldsymbol{m}'$ such that $\boldsymbol{l}+\boldsymbol{l}' \neq \boldsymbol{m}+\boldsymbol{m}'$.

Two applications are shown in this paper. The former is used for dynamic Huffman tree. Consider a weight $\boldsymbol{w}$ and a level vector $\boldsymbol{l}$ in $\mathcal{L}_n$ such that $\boldsymbol{l}$ is optimal under $\boldsymbol{w}$. For this vector, there are $O(Ln^3)$ similar level vectors, where L is the most largest level of $T(\boldsymbol{l})$. Since each optimal region of similar level vector and $R(\boldsymbol{l})$ share a point, there exist only these optimal regions around the $R(\boldsymbol{l})$. If optimal tree $T(\boldsymbol{l})$ is transformed to other extended binary tree when $\boldsymbol{w}$ is slightly changed, the optimal tree is included in $T(\boldsymbol{l}')$ where $\boldsymbol{l}'$ is similar level vector of $\boldsymbol{l}$. It is possible that the number of considerable similar tree is smaller than $O(Ln^3)$ for a given weight $\boldsymbol{w}$.

It is consider that conv($\mathcal{L}_n$) has a relation with permuto-associahedron $K\Pi_{n-1}$([5, p.19]). The polytope is defined by the combination of permutahedron and associahedron. Each vertex of $K\Pi_{n-1}$ is a way of multiplication of n terms without commutative law and associative law. For example, $K\Pi_2$ has 12 vertices (see Figure 5). The vertex also corresponds to a labeled extended binary tree with n leaves. Level vector is defined for labeled extended binary tree. So, a projection from vertices $K\Pi_{n-1}$ to $\mathcal{L}_n$ is defined. In Figure 5, the vertices on $K\Pi_2$ by italic font map to a vertex $(2,1,2)$ on conv($\mathcal{L}_n$).

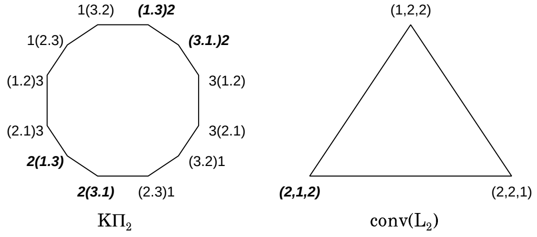


Figure 5: permuta-associahedron $K\Pi_2$ (left) and conv($\mathcal{L}_3$) (right)

The following conjecture is considered:

Conjecture 1 *If there is an edge between two vertices of $K\Pi_{n-1}$, corresponding vertices of $\mathcal{L}_n$ are one vertex or connected by an edge on conv($\mathcal{L}_n$).*

Finally, we state about the relation between optimal region of Huffman tree and that of alphabetic tree for *one* level vector. *Alphabetic tree* is also extended binary tree such that the labels of leaves appear in alphabetic order from left to right if we traverse the tree in inorder. Level vector of alphabetic tree is similarly defined with that of Huffman. Let $\mathcal{A}_n$ be the set of all level vectors of alphabetic tree with n leaves. The definitions of $\mathcal{A}_n$ is similar to that of $\mathcal{L}_n$. Moreover , for $\mathcal{A}_n$ similar results of this paper can be shown. The results will be appeared in the other paper. In this paper, we state a relation between optimal regions $R(\boldsymbol{l})$ in $\mathcal{L}_n$ and $S(\boldsymbol{l})$ in $\mathcal{A}_n$. Let $\boldsymbol{l}$ be a level vector in $\mathcal{A}_n$. This vector is also included in $\mathcal{L}_n$. Any point in $\mathcal{L}_n$ is optimal in $\mathcal{A}_n$, but opposite is not. Then the following relation is shown:

$$R(\boldsymbol{l}) \subsetneq S(\boldsymbol{l}).$$

In information theory it is shown that the average length of code by Huffman tree becomes optimal and by alphabetic tree is equal to optimal plus extra. This fact is suitable for the relation above.

Acknowledgements

The author would like to thank Prof. Tomomi Matsui, Prof. David Avis and Prof. Luc Devroye for useful discussions and pointing out several errors. The author also thanks Prof. Mamoru Hoshi and Prof. Hiroshi Imai for useful suggestions.

References

[1] D. Avis: Living with lrs, *Lecture Notes in Computer Science*, Vol. 1763, 2000, pp. 47-56.

[2] D.E. Knuth: *The Art of Computer Programming*, Vol. 1, Third Edition, Addison-Wesley, 1997.

[3] D.E. Knuth: *The Art of Computer Programming*, Vol. 3, Second Edition, Addison-Wesley, 1998.

[4] S. V. Nagaraj: Optimal binary search trees, *Theoretical Computer Science*, no. 188, 1997, pp. 1 - 44.

[5] G. M. Ziegler: *Lectures on Polytopes*, Springer-Verlag, 1995.