Proof of Optimality of Huffman Codes

CSC373 Spring 2009

1 Problem
You are given an alphabet $A$ and a frequency function $f : A \rightarrow (0, 1)$ such that $\sum_x f(x) = 1$. Find a binary tree $T$ with $|A|$ leaves (each leaf corresponding to a unique symbol) that minimizes

$$\text{ABL}(T) = \sum_{\text{leaves of } T} f(x) \text{depth}(x)$$

Such a tree is called optimal.

2 Algorithm

HUF($A, f$)
If $|A| = 1$ then return a single vertex.
Let $w$ and $y$ be the symbols with the lowest frequencies.
Let $A' = A \setminus \{w, y\} + \{z\}$.
Let $f'(x) = f(x)$ for all $x \in A' \setminus \{z\}$, and let $f'(z) = f(w) + f(y)$.
$T' = \text{HUF}(A', f')$.
Create $T$ from $T'$ by adding $w$ and $y$ as children of $z$.
return $T$

3 Proof

Lemma 1 Let $T$ be a tree for some $f$ and $A$, and let $y$ and $w$ be two leaves. Let $T'$ be the tree obtained from $T$ by swapping $y$ and $w$. Then $\text{ABL}(T') - \text{ABL}(T) = (f(y) - f(w))(\text{depth}(w, T) - \text{depth}(y, T))$.

Proof

$$\text{ABL}(T') - \text{ABL}(T) = f(y)\text{depth}(w, T) + f(w)\text{depth}(y, T) - f(w)\text{depth}(w, T) - f(y)\text{depth}(y, T)$$
$$= f(y)(\text{depth}(w, T) - \text{depth}(y, T)) + f(w)(\text{depth}(y, T) - \text{depth}(w, T))$$
$$= (f(y) - f(w))(\text{depth}(w, T) - \text{depth}(y, T))$$

Lemma 2 There exists an optimal tree such that the two symbols with the lowest frequencies are siblings.

Proof Let $T$ be an optimal tree. Let $w$ and $y$ be two symbols with the lowest frequencies. If there is more than one symbol that has the lowest frequency, then
take two that have the biggest depth. If \( w \) and \( y \) are siblings, then we are done. Otherwise, suppose without loss of generality, that \( \text{depth}(w, T) \geq \text{depth}(y, T) \).

We have three cases:

- **\( w \)** has a sibling \( z \). Let \( T' \) be the tree created from \( T \) by swapping \( z \) and \( y \), and thus making \( w \) and \( y \) siblings. By applying Lemma 1, we get that \( \text{ABL}(T') \leq \text{ABL}(T) \). Since \( T \) is optimal, there cannot be another tree with a smaller cost, and so \( \text{ABL}(T') = \text{ABL}(T) \). Thus \( T' \) is also optimal.

- **\( w \)** is an only child. Create \( T' \) by removing \( w \)'s leaf and assigning \( w \) to its old parent. \( T' \) is cheaper than \( T \), contradiction the optimality of \( T \). Hence, this case is not possible.

- There exists a node \( z \) at a depth bigger than \( w \). Create \( T' \) by swapping \( w \) and \( z \). By our choice of \( w \), \( f(w) < f(z) \), so, applying Lemma 1, we have that \( T' \) is cheaper than \( T \), a contradiction. Hence, this case is not possible.

**Theorem 3** The algorithm \( \text{HUF}(A, f) \) computes an optimal tree for frequencies \( f \) and alphabet \( A \).

**Proof** The proof is by induction on the size of the alphabet. The induction hypothesis is that for all \( A \) with \( |A| = n \) and for all frequencies \( f \), \( \text{HUF}(A, f) \) computes the optimal tree.

In the base case \((n = 1)\), the tree is only one vertex and the cost is zero, which is the smallest possible.

For the general case, assume that the induction hypothesis holds for \( n - 1 \). That is, \( T' \) is optimal for \( A' \) and \( f' \). First, let us show the following:

\[
\text{ABL}(T) = \left( \sum_{x \in A \setminus \{w, y\}} f(x) \text{depth}(x, T) + f(w) \text{depth}(w, T) + f(y) \text{depth}(y, T) \right)
\]

\[
= \left( \sum_{x \in A \setminus \{w, y\}} f(x) \text{depth}(x, T) + (f(w) + f(y))(\text{depth}(z, T') + 1) \right)
\]

\[
= \left( \sum_{x \in A \setminus \{w, y\}} f(x) \text{depth}(x, T) + f'(z) \text{depth}(z, T') + f(w) + f(y) \right)
\]

\[
= \left( \sum_{x \in A'} f'(x) \text{depth}(x, T') + f(w) + f(y) \right)
\]

\[
= \text{ABL}(T') + f(w) + f(y)
\]

Now, assume for the sake of contradiction that \( T \) is not optimal, and let \( Z \) be an optimal tree that has \( w \) and \( y \) as siblings (this exists by the above lemma).

Let \( Z' \) be the tree obtained from \( Z \) by removing \( w \) and \( y \). We can view \( Z' \) as a tree for the alphabet \( A' \) and frequency function \( f' \). We can then repeat the calculation above and get \( \text{ABL}(Z) = \text{ABL}(Z') + f(w) + f(y) \). So, \( \text{ABL}(T') = \text{ABL}(T) - f(w) - f(y) > \text{ABL}(Z) - f(w) - f(y) = \text{ABL}(Z') \). Since \( T' \) is optimal for \( A' \) and \( f' \), this is a contradiction.