Huffman Codes

Correctness Proof

Algorithms: Design and Analysis, Part II

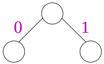
Correctness of Huffman's Algorithm

Theorem: [Huffman 52] Huffman's algorithm computes a binary tree (with leaves $\leftrightarrow$ symbols of Σ) that minimizes the average encoding length

$$L(T) = \sum_{i \in \Sigma} p_i [\text{depth of leaf } i \text{ in } T].$$

Proof: By induction on $n = |\Sigma|$. (Can assume $n \geq 2$.)

Base case: When $n = 2$, algorithm outputs the optimal tree.
(Needs 1 bit per symbol)



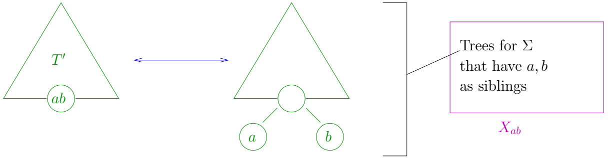
Inductive step: Fix input with $n = |\Sigma| > 2$.

By inductve hypothesis: Algorithm solves smaller subproblems (for Σ') optimally.

Inductive Step

Let $\Sigma' = \Sigma$ with a, b (symbols with smallest frequencies) replaced by meta-symbol ab. Define $p_{ab} = p_a + p_b$.

Recall: Exact correspondence between:



Important: For every such pair T' and T, $L(T) - L(T')$ is (after cancellation)

p_a [a's depth in T] $+ p_b$ [b's depth in T] $- p_{ab}$ [ab's depth in T'] $=$

Each is one more than

$$= p_A(d+1) + p_b(d+1) - (p_a + p_b)d = p_a + p_b,$$ Independent of T, T'!

Proof of Theorem

Inductive hypothesis: Huffman's algorithm computes a tree $\hat{T}'$ that minimizes $L(T')$ for Σ'.

Upshot of last slide: Corresponding tree $\hat{T}$ minimizes $L(T)$ for Σ over all trees in X_{ab} (i.e., where a & b are siblings)

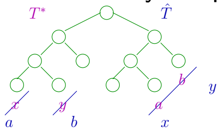
Key lemma: [Completes proof of theorem] There is an optimal tree (for Σ) in X_{ab}. [i.e., a & b were "safe" to merge]

Intuition: Can make an optimal tree better by pushing a & b as deep as possible (since a, b have smallest frequencies).

Proof of Key Lemma

By exchange argument. Let T^* be any tree that minimizes $L(T)$ for Σ. Let x, y be siblings at the deepest level of T^*.

The exchange: Obtain $\hat{T}$ from T^* by swapping $a \leftrightarrow x$, $b \leftrightarrow y$



Note: $\hat{T} \in X_{ab}$ (by choice of x, y).

To finish: Will show that $L(\hat{T}) \leq L(T^*)$
[$\Rightarrow$ $\hat{T}$ also optimal, completes proof]

Reason:

$$\begin{aligned} L(T^*) - L(\hat{T}) &= (p_x - p_a) \quad [x\text{'s depth in } T^* \text{ - } a\text{'s depth in } T^*] \\ &+ (p_y - p_b) \quad [y\text{'s depth in } T^* \text{ - } b\text{'s depth in } T^*] \\ &\geq 0 \quad \text{QED!} \end{aligned}$$

≥ 0 since a, b have smallest frequencies $\qquad$ ≥ 0 by choice of x, y

Notes on Running Time

Naive implementation: $O(n^2)$ time, where $n = |\Sigma|$.

Speed ups: - Use a heap! [to perform repeated minimum computations]
- Use keys = frequencies
- After extracting the two smallest-frequency symbols, re-Insert the new meta-symbol [new key = sum of the 2 old ones]
$\Rightarrow$ Iterative, $O(n \log n)$ implementation.

Even faster: (Non-trivial exercise) Sorting + $O(n)$ additional work.
- Manage (meta-)symbols using two queues.