INFORMATION THEORY & CODING

Part 7 : Source Coding 3 - Huffman Code

Dr. Rui Wang

Department of Electrical and Electronic Engineering
Southern Univ. of Science and Technology (SUSTech)

Email: wang.r@sustech.edu.cn

November 7, 2023

Huffman Codes

Problem 5.1

Given source symbols and their probabilities of occurence, how to design an optimal source code (prefix code and the shortest on average)?

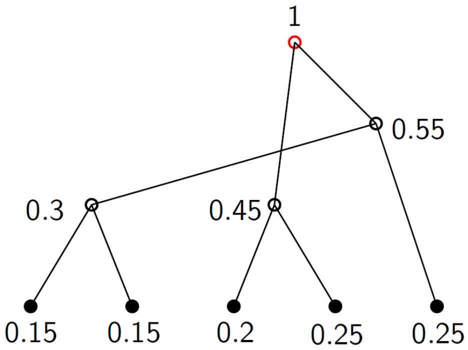
Huffman Codes

- Merge the D symbols with the smallest probabilities, and generate one new symbol whose probability is the summation of the D smallest probabilities.
- Assign the D corresponding symbols with digits $0, 1, \ldots, D-1$, then go back to Step 1.

Repeat the above process until D probabilities are merged into probability 1.

Example 1

x	p(x)
1	0.25
2	0.25
3	0.2
4	0.15
5	0.15



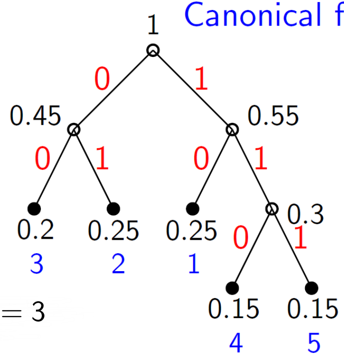
Reconstruct the tree

Huffman Codes: A few examples

Example 1

x	p(x)	C(x)
1	0.25	10
2	0.25	01
3	0.2	00
4	0.15	110
5	0.15	111

Canonical form



Validations:

$$\ell(1) = \ell(2) = \ell(3) = 2, \ell(4) = \ell(5) = 3$$

$$\text{Ł} = \sum \ell(x)p(x) = 2.3\text{bits}$$

$$H_2(X) = -\sum p(x)\log_2 p(x) = 2.29\text{bits}$$

$$L \geq H_2(X)$$

Huffman Codes: A few examples

Example 2

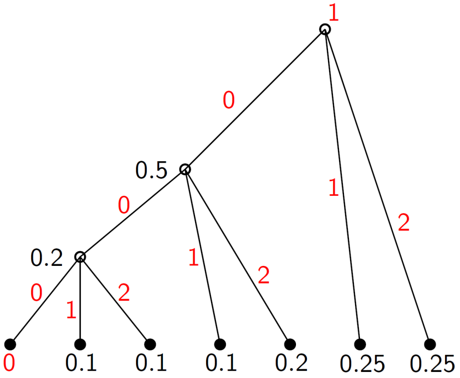
x	p(x)
1	0.25
2	0.25
3	0.2
4	0.1
5	0.1
6	0.1
Dummy	0

$\mathcal{D} = \{0, 1, 2\}$

At one time, we merge D symbols, and at each stage of the reduction, the number of symbols is reduced by $D - 1$. We want the total # of symbols to be $1 + k(D - 1)$. If not, we add dummy symbols with probability 0.

Example 2 ($D \geq 3$)

x	p(x)	C(x)
1	0.25	1
2	0.25	2
3	0.2	02
4	0.1	01
5	0.1	002
6	0.1	001
Dummy	0	000



Validations:

$L = \sum \ell(x)p(x) = 1.7$ ternary digits

$H_3(X) = -\sum p(x) \log_3 p(x) \approx 1.55$ ternary digits

Optimality of Huffman Codes

Lemma 5.8.1

For any distribution, the optimal prefix codes (with minimum expected length) should satisfy the following properties:

1. *If $p_j > p_k$, then $\ell_j \leq \ell_k$.*
2. *The two longest codewords have the same length.*
3. *There exists an optimal prefix code, such that two of the longest codewords differ only in the last bit and correspond to the two least likely symbols.*

Optimality of Huffman Codes

- 1. If $p_j > p_k$, then $\ell_j \leq \ell_k$.

Proof.

Suppose that C_m is an optimal code. Consider C'_m, with the codewords j and k of C_m interchanged. Then

$$\underbrace{L\left(C'_m\right) - L\left(C_m\right)}_{\geq 0} = \sum p_i \ell'_i - \sum p_i \ell_i$$

$$= p_j \ell_k + p_k \ell_j - p_j \ell_j - p_k \ell_k$$

$$= \underbrace{\left(p_j - p_k\right)}_{>0} \left(\ell_k - \ell_j\right)$$

Thus, we must have $\ell_k \geq \ell_j$. □

Optimality of Huffman Codes

- 2. The two longest codewords have the same length.

Optimality of Huffman Codes

- 3. There exists an optimal prefix code, such that two of the longest codewords differ only in the last bit and correspond to the two least likely symbols.

Proof.

If there is a maximal-length codeword without a sibling, we can delete the last bit of the codeword and still preserve the prefix property. This reduces the average codeword length and contradicts the optimality of the code. Hence, every maximum-length codeword in any optimal code has a sibling. Now we can exchange the longest codewords s.t. the two lowest-probability source symbols are associated with two siblings on the tree, without changing the expected length. □

Optimality of Huffman Codes

Lemma 5.8.1

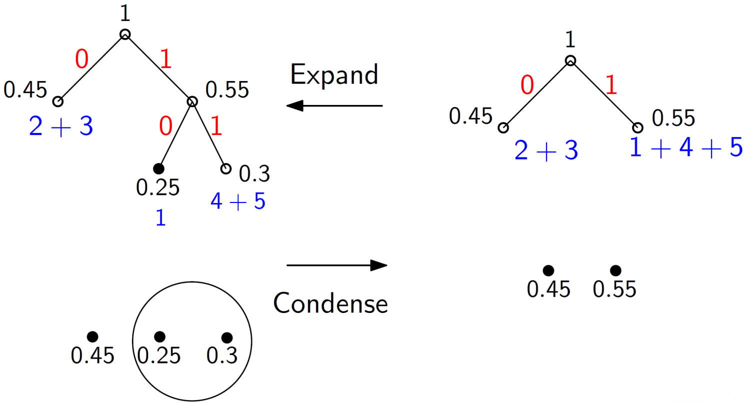
For any distribution, the optimal prefix codes (with minimum expected length) should satisfy the following properties:

1. *If $p_j > p_k$, then $\ell_j \leq \ell_k$.*
2. *The two longest codewords have the same length.*
3. *There exists an optimal prefix code, such that two of the longest codewords differ only in the last bit and correspond to the two least likely symbols.*

$\Rightarrow$ If $p_1 \geq p_2 \geq \cdots p_m$, then there exists an optimal code with $\ell_1 \leq \ell_2 \leq \cdots \ell_{m-1} = \ell_m$, and codewords $C(x_{m-1})$ and $C(x_m)$ differ only in the last bit.
(canonical codes)

Optimality of Huffman Codes

- We prove the optimality of Huffman codes by induction. Assume binary code in the proof.



Optimality of Huffman Codes

Proof.

For $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define the Huffman reduction $\mathbf{p}' = (p_1, p_2, \ldots, p_{m-1+p_m})$ over an alphabet size of $m-1$. Let $C^*_{m-1}(\mathbf{P}')$ be an optimal Huffman code for $\mathbf{p}'$, and let $C^*_m(\mathbf{p})$ be the canonical optimal code for $\mathbf{p}$. □

Key idea.

expand C^*_{m-1} to $C_m(\mathbf{p}) \Rightarrow L(C_m) = L(C^*_m)$

Optimality of Huffman Codes

Proof.

For $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define the Huffman reduction $\mathbf{p}' = (p_1, p_2, \ldots, p_{m-1+p_m})$ over an alphabet size of $m-1$. Let $C^*_{m-1}(\mathbf{P}')$ be an optimal Huffman code for $\mathbf{p}'$, and let $C^*_m(\mathbf{p})$ be the canonical optimal code for $\mathbf{p}$. □

	$C^*_{m-1}(\mathbf{p}')$		$C_m(\mathbf{p})$	
p_1	w'_1	l'_1	$w_1 = w'_1$	$l_1 = l'_1$
p_2	w'_2	l'_2	$w_2 = w'_2$	$l_2 = l'_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_{m-2}	w'_{m-2}	l'_{m-2}	$w_{m-2} = w'_{m-2}$	$l_{m-2} = l'_{m-2}$
$p_{m-1} + p_m$	w'_{m-1}	l'_{m-1}	$w_{m-1} = w'_{m-1}0$	$l_{m-1} = l'_{m-1} + 1$
			$w_m = w'_{m-1}1$	$l_m = l'_{m-1} + 1$

Optimality of Huffman Codes

Proof.

For $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define the Huffman reduction $\mathbf{p}' = (p_1, p_2, \ldots, p_{m-1+p_m})$ over an alphabet size of $m-1$. Let $C^*_{m-1}(\mathbf{P}')$ be an optimal Huffman code for $\mathbf{p}'$, and let $C^*_m(\mathbf{p})$ be the canonical optimal code for $\mathbf{p}$. □

$C_{m-1}(\mathbf{p}')$			$C^*_m(\mathbf{p})$	
p_1	w'_1	l'_1	$w_1 = w'_1$	$l_1 = l'_1$
p_2	w'_2	l'_2	$w_2 = w'_2$	$l_2 = l'_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
p_{m-2}	w'_{m-2}	l'_{m-2}	$w_{m-2} = w'_{m-2}$	$l_{m-2} = l'_{m-2}$
$p_{m-1} + p_m$	w'_{m-1}	l'_{m-1}	$w_{m-1} = w'_{m-1}0$	$l_{m-1} = l'_{m-1} + 1$
			$w_m = w'_{m-1}1$	$l_m = l'_{m-1} + 1$

Optimality of Huffman Codes

Proof.

For $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define the Huffman reduction $\mathbf{p}' = (p_1, p_2, \ldots, p_{m-1+p_m})$ over an alphabet size of $m-1$. Let $C^*_{m-1}(\mathbf{P}')$ be an optimal Huffman code for $\mathbf{p}'$, and let $C^*_m(\mathbf{p})$ be the canonical optimal code for $\mathbf{p}$. □

expand $C^*_{m-1}(\mathbf{p}')$ to $C_m(\mathbf{p})$

$$L(\mathbf{p}) = L^*(\mathbf{p}') + p_{m-1} + p_m$$

condense $C^*_m(\mathbf{p})$ to $C_{m-1}(\mathbf{p}')$

$$L^*(\mathbf{p}) = L(\mathbf{p}') + p_{m-1} + p_m$$

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Optimality of Huffman Codes

Proof.

For $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define the Huffman reduction $\mathbf{p}' = (p_1, p_2, \ldots, p_{m-1+p_m})$ over an alphabet size of $m-1$. Let $C^*_{m-1}(\mathbf{P}')$ be an optimal Huffman code for $\mathbf{p}'$, and let $C^*_m(\mathbf{p})$ be the canonical optimal code for $\mathbf{p}$. □

$$L(\mathbf{p}) = L^*\left(\mathbf{p}'\right) + p_{m-1} + p_m$$
$$L^*(\mathbf{p}) = L\left(\mathbf{p}'\right) + p_{m-1} + p_m$$

$$\underbrace{\left(L\left(\mathbf{p}'\right) - L^*\left(\mathbf{p}'\right)\right)}_{\geq 0} + \underbrace{\left(L(\mathbf{p}) - L^*(\mathbf{p})\right)}_{\geq 0} = 0$$

Optimality of Huffman Codes

Proof.

For $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define the Huffman reduction $\mathbf{p}' = (p_1, p_2, \ldots, p_{m-1+p_m})$ over an alphabet size of $m-1$. Let $C^*_{m-1}(\mathbf{P}')$ be an optimal Huffman code for $\mathbf{p}'$, and let $C^*_m(\mathbf{p})$ be the canonical optimal code for $\mathbf{p}$. □

Thus, $L(\mathbf{p}) = L^*(\mathbf{p})$. Minimizing the expected length $L(C_m)$ is equivalent to minimizing $L(C_{m-1})$. The problem is reduced to one with $m-1$ symbols and probability masses $(p_1, p_2, \ldots, p_{m-1} + p_m)$. Proceeding this way, we finally reduce the problem to two symbols, in which case the optimal code is obvious.

Optimality of Huffman Codes

Theorem 5.8.1

Huffman coding is optimal, that is, if C^* is a Huffman code and C' is any other uniquely decodable code, $L(C^*) \leq L(C')$.

Remark

Huffman coding is a *greedy algorithm* in which it merges the two least likely symbols at each step.

LOCAL OPT $\rightarrow$ GLOBAL OPT

Textbook

Related Sections : 5.6 - 5.8