# INFORMATION THEORY & CODING

## Week 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 8, 2022

## 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.

Huffman code:

Step I: make sure: $1+(D-1)k$ symbols

Step II: D symbols with the least probabilities

$\Rightarrow$ one symbol with sum probabilities.

Step III: repeat step II until one symbol left.

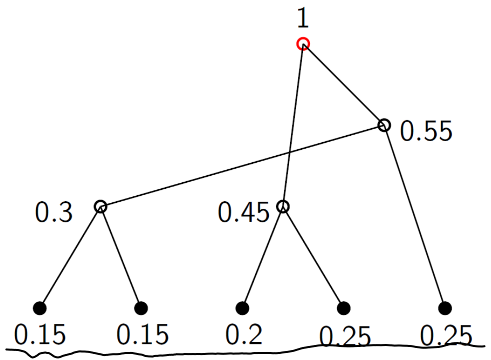
Step IV: Assign codeword.

# Huffman Codes: A few examples

**Example 1**

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

从下往上看

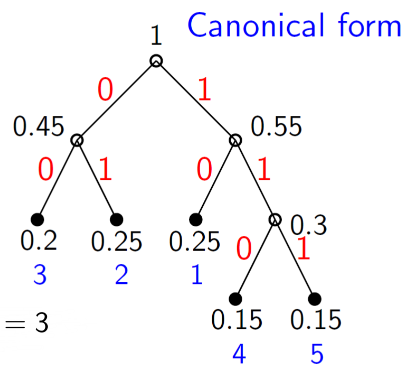



从小到大

Reconstruct the tree

**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 |




**Validations:**

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

Ł $= \sum \ell(x)p(x) = 2.3$bits

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

不取等号的原因：分布不遵循 $D^{-l_i}$

$L \geq H_2(X)$

**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 |

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

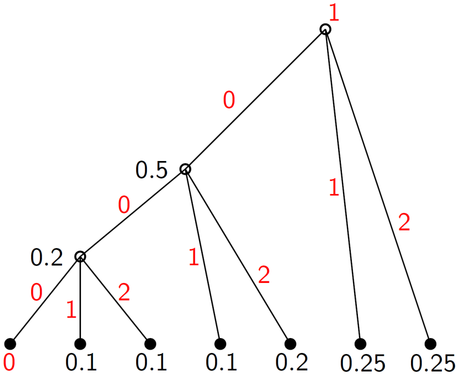
At one time, we merge $D$ symbols, and at each stage of the reduction, the number of symbols is reduced by $D - 1$. We want the total # of symbols to be $1 + k(D - 1)$. If not, we add dummy symbols with probability 0.

positive integer

D-ary

**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.*

proof: Suppose $C$ with $p_j > p_k$ and $l_j > l_k$.
we should prove $C$ is not optimal.

$$L(C) = \sum_i p_i l_i = p_j l_j + p_k l_k + \sum_{i \neq j,k} p_i l_i$$

$C'$: $j \to l_k$; $k \to l_j$; $C$.

$$L(C') = p_j l_k + p_k l_j + \sum_{i \neq j,k} p_i l_i$$

$$L(C) - L(C') = \underbrace{(p_j - p_k)}_{>0} \underbrace{(l_j - l_k)}_{>0} > 0$$

$L(C) > L(C') \Rightarrow C$ is not optimal.

proof: code $C$.

$C(j)$: $b_1 b_2 \cdots b_m$

$C(k)$: $b_1' b_2' \cdots b_m' b_{m+1}' \cdots b_{m+n}'$

$C'$

$$C'(i) = \begin{cases} C(i), & i \neq k \\ \underbrace{b_1' b_2' \cdots b_m'}, & i = k \end{cases} \Rightarrow C' \text{ is prefix}$$

$\Rightarrow L(C') < L(C)$

不会为$C'$中其它codeword的prefix

and. $C'$中其它codeword不会为它的prefix

# 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

$$\begin{aligned}
\underbrace{L\left(C'_m\right) - L\left(C_m\right)}_{\geq 0} &= \sum p_i \ell'_i - \sum p_i \ell_i \\
&= p_j \ell_k + p_k \ell_j - p_j \ell_j - p_k \ell_k \\
&= \underbrace{\left(p_j - p_k\right)}_{>0} \left(\ell_k - \ell_j\right)
\end{aligned}$$

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 maximal-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. □

proof: Optimal prefix code $C$.

longest codeword. length $= m$.

$b_1 b_2 \cdots\cdots b_m \in C$

$b_1' b_2' \cdots\cdots b_m' \in C$.

consider a new codeword. $b_1 b_2 \cdots\cdots b_{m-1} \overline{b_m}$

① $b_1 b_2 \cdots\cdots b_{m-1} \overline{b_m} \in C$. if $\checkmark \Rightarrow C$ is we wanted

② $b_1 b_2 \cdots\cdots b_{m-1} \overline{b_m} \notin C$

If ② is true, construct $C'$: replace $b_1' b_2' \cdots b_m'$ of $C$ by $b_1 b_2 \cdots b_{m-1} \overline{b_m}$.

$\Downarrow$

optimal prefix code

$\rightarrow$ canonical form.

# Optimality of Huffman Codes

## Lemma 5.8.1

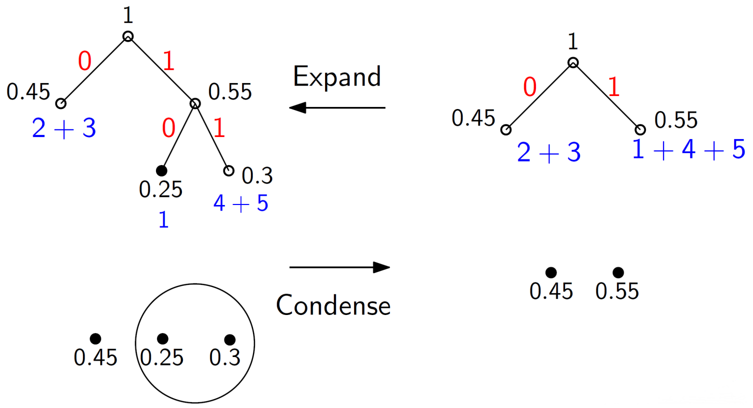
*For any distribution, the optimal prefix codes (with minimum expected length) should satisfy the following properties:*

1. *If $p_j > p_k$, then $\ell_j \le \ell_k$.*
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 $p_1 \ge p_2 \ge \cdots p_m$, then there exists an optimal code with $\ell_1 \le \ell_2 \le \cdots \ell_{m-1} = \ell_m$, 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.






$p_1, p_2 \cdots p_{m+1}$

$p_1 \geq p_2 \geq \cdots \geq p_{m+1}$

Huffman code for $(p_1 p_2 \cdots p_{m+1})$

$\Downarrow$ condense

Huffman code for $(p_1, p_2 \cdots p_{m-1}, p_m + p_{m+1})$.

prefix code of C-form for $(p_1 p_2 \cdots p_{m+1})$

$\Downarrow$ condense.

prefix code for $(p_1, p_2 \cdots p_{m-1}, p_m + p_{m+1})$

W.L.O.G 2-ary code.

Step I: Huffman code is optimal for two-symbol dist $|\mathcal{X}| = 2$.

Step II: Suppose Huffman code is optimal for $|\mathcal{X}| = m$, $m \geq 2$.

Step III: We should prove: Huffman code is optimal for $|\mathcal{X}| = m+1$, $p_1 \geq p_2 \geq \cdots \geq p_{m+1}$

$C_{m+1}$: Huffman code for $(p_1, p_2 \cdots p_m, p_{m+1})$

$C_m^*$: code condensed from $C_{m+1}$

$\Rightarrow$ Huffman code for $(p_1, p_2, \cdots, p_m + p_{m+1})$

$\downarrow$

根据 Step II. it is optimal.

$C_{m+1}^*$: optimal prefix code of C-form for $(p_1, p_2, \cdots p_{m+1})$

$C_m$: condensed from $C_{m+1}^* \Rightarrow$ prefix code for $(p_1, p_2, \cdots p_m + p_{m+1})$

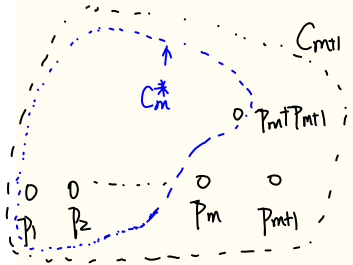
## Optimality of Huffman Codes

**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 $\mathbf{p}$. □

**Key idea.**

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




$L(C_{m+1}) = \sum_{i=1}^{m+1} p_i l_i = \sum_{i=1}^{m-1} p_i l_i + p_m \underline{l_m} + p_{m+1} \underline{l_{m+1}}$

$l_m = l_{m+1}$

$L(C^*_m) = \sum_{i=1}^{m-1} p_i l_i + (p_m + p_{m+1})(l_{m+1} - 1)$

$L(C_{m+1}) - L(C^*_m) = p_m + p_{m+1}$

$L(C^*_{m+1}) - L(C_m) = p_m + p_{m+1}$

$\underbrace{L(C_{m+1}) - L(C^*_{m+1})}_{\geq 0} = \underbrace{L(C^*_m) - L(C_m)}_{\leq 0}$

$\rightarrow L(C_{m+1}) = L(C^*_{m+1}) = L(C^*_m) = L(C_m)$

# 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 $\mathbf{p}$. □

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

# Optimality of Huffman Codes

**Proof.**

For $\mathbf{p} = (p_1, p_2, \dots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define the Huffman reduction $\mathbf{p}' = (p_1, p_2, \dots, p_{m-1+p_m})$ 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 $\mathbf{p}$. □

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

**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 $C_{m-1}(\mathbf{p}')$

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

# 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 $\mathbf{p}$. $\square$

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.

### 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