INFORMATION THEORY & CODING

Week 6 : Source Coding 2

Dr. Rui Wang

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

Email: wang.r@sustech.edu.cn

October 20, 2020

Review Summary

- **Classes of codes**

  Prefix codes $\Rightarrow$ Uniquely decodable codes $\Rightarrow$ Nonsingular codes

- **Kraft inequality**

  Prefix codes $\Leftrightarrow \sum D^{-\ell_i} \leq 1$.

Outline

- **Kraft inequality for uniquely decodable code**

  Uniquely decodable code does NOT provide more choices than prefix code

- **Bounds on optimal expected length**

  Entropy length is achievable when jointly encoding a random sequence.

- **Huffman Code:** algorithm to find the optimal code with shortest expected length

Kraft Inequaltiy for Uniquely Decodable Codes

Theorem 5.5.1 (McMillan)

The codeword lengths of any uniquely decodable D-ary code must satisfy the Kraft inequality

$$\sum D^{-\ell_i} \leq 1.$$

Conversely, given a set of codeword lengths that satisfy this inequality, it is possible to construct a uniquely decodable code with these codeword lengths.

Proof.

Consider C^k, the k-th extension of the code by k repetitions. Let the codeword lengths of the symbols $x \in \mathcal{X}$ be $\ell(x)$. For the k-th extension code, we have

$$\ell(x_1, x_2, \ldots, x_k) = \sum_{i}^{k} \ell(x_i).$$

□

Kraft Inequaltiy for Uniquely Decodable Codes

Theorem 5.5.1 (McMillan)

The codeword lengths of any uniquely decodable D-ary code must satisfy the Kraft inequality

$$\sum D^{-\ell_i} \leq 1.$$

Proof. (cont.)

Consider

$$\left(\sum_{x\in\mathcal{X}} D^{-\ell(x)}\right)^k = \sum_{x_1\in\mathcal{X}}\sum_{x_2\in\mathcal{X}}\cdots\sum_{x_k\in\mathcal{X}} D^{-\ell(x_1)}D^{-\ell(x_2)}\ldots D^{-\ell(x_k)}$$

$$= \sum_{x_1,x_2,\cdots x_k\in\mathcal{X}^k} D^{-\ell(x_1)}D^{-\ell(x_2)}\ldots D^{-\ell(x_k)}$$

$$= \sum_{x^k\in\mathcal{X}^k} D^{-\ell(x^k)}$$

Kraft Inequaltiy for Uniquely Decodable Codes

Theorem 5.5.1 (McMillan)

The codeword lengths of any uniquely decodable D-ary code must satisfy the Kraft inequality

$$\sum D^{-\ell_i} \leq 1.$$

Proof. (cont.)

Let $\ell_{\max}$ be the maximum codeword length and $a(m)$ is the number of source sequences x^k mapping into codewords of length m. Unique decodability implies that $a(m) \leq D^m$. We have

$$\begin{aligned}\Big(\sum_{x\in\mathcal{X}} D^{-\ell(x)}\Big)^k = \sum_{x^k\in\mathcal{X}^k} D^{-\ell(x^k)} &= \sum_{m=1}^{k\ell_{\max}} a(m)D^{-m} \\ &\leq \sum_{m=1}^{k\ell_{\max}} D^m D^{-m} \\ &= k\ell_{\max}\end{aligned}$$

Kraft Inequaltiy for Uniquely Decodable Codes

Theorem 5.5.1 (McMillan)

The codeword lengths of any uniquely decodable D-ary code must satisfy the Kraft inequality

$$\sum D^{-\ell_i} \leq 1.$$

Proof. (cont.)

$$\left(\sum_{x \in \mathcal{X}} D^{-\ell(x)} \right)^k \leq k\ell_{\max}.$$

Hence,

$$\sum_j D^{-\ell_j} \leq (k\ell_{\max})^{1/k}$$

holds for all k. Since the RHS$\rightarrow 1$ as $k \rightarrow \infty$, we prove the Kraft inequality. For the converse part, we can construct a prefix code as in **Theorem 5.2.1**, which is also uniquely decodable. □

Optimal Codes

Problem To find the set of lengths $\ell_1, \ell_2, \ldots, \ell_m$ satisfying the Kraft inequality and whose expected length $L = \sum p_i \ell_i$ is minimized.

Optimization:

minimize $L = \sum p_i \ell_i$

subject to $\sum D^{-\ell_i} \leq 1$ and ℓ_i's are integers.

Optimal Codes

Theorem 5.3.1

The expected length L of any prefix D-ary code for a random variable X is no less than $H_D(X)$, i.e.,

$$L \geq H_D(X),$$

with equality iff $D^{-\ell_i} = p_i$.

Proof.

$$\begin{aligned} L - H_D(X) &= \sum p_i \ell_i - \sum p_i \log_D \frac{1}{p_i} \\ &= -\sum p_i \log_D D^{-\ell_i} + \sum p_i \log_D p_i \\ &= \sum p_i \log_D \frac{p_i}{r_i} - \log_D c \\ &= D(\mathbf{p} \| \mathbf{r}) + \log_D \frac{1}{c} \geq 0 \end{aligned}$$

"=" holds if $c = 1$ and $r_i = p_i$.

where $r_i = D^{-\ell_i} / \sum_j D^{\ell_j}$ and $c = \sum D^{-\ell_i} \leq 1$. □

Optimal Codes

Theorem 5.3.1

The expected length L of any prefix D-ary code for a random variable X is no less than $H_D(X)$, i.e.,

$$L \geq H_D(X),$$

with equality iff $D^{-\ell_i} = p_i$.

Definition

A probability distribution is called D-adic if each of the probabilities is equal to D^{-n} for some n. Thus, we have equality in the theorem iff the distribution of X is D-adic.

Remark

$H_D(X)$ is a lower bound on the optimal code length. The equality holds iff p is D-adic.

Bound on the Optimal Code Length

Theorem 5.4.1 (Shannon Codes)

Let $\ell_1^*, \ell_2^*, \ldots, \ell_m^*$ be optimal codeword lengths for a source distribution $\mathbf{p}$ and a D-ary alphabet, and let L^* be the associated expected length of an optimal code ($L^* = \sum p_i \ell_i^*$). Then

$$H_D(X) \le L^* < H_D(X) + 1.$$

Proof.

Take $\ell_i = \lceil -\log_D p_i \rceil$. Since

$$\sum_{i \in \mathcal{X}} D^{-\ell_i} \le \sum p_i = 1,$$

these lengths satisfy Kraft inequality and we can create a prefix code. Thus,

$$\begin{aligned} L^* &\le \sum p_i \lceil -\log_D p_i \rceil \\ &< \sum p_i(-\log_D p_i + 1) \\ &= H_D(X) + 1. \end{aligned}$$

□

Bound on the Optimal Code Length

Theorem 5.4.2

Consider a system in which we send a sequence of n symbols from X. The symbols are assumed to be i.i.d. according to $p(x)$. The minimum expected codeword length per symbol satisfies

$$\frac{H(X_1, X_2, \ldots, X_n)}{n} \leq L_n^* < \frac{H(X_1, X_2, \ldots, X_n)}{n} + \frac{1}{n}.$$

Proof.

First,

$$L_n = \frac{1}{n} \sum p(x_1, x_2, \ldots, x_n) \ell(x_1, x_2, \ldots, x_n) = \frac{1}{n} E[\ell(X_1, X_2, \ldots, X_n)]$$

We also have

$$H(X_1, X_2, \ldots, X_n) \leq E[\ell(X_1, X_2, \ldots, X_n)] < H(X_1, X_2, \ldots, X_n) + 1.$$

Since $X_1, X_2, \ldots, X_n$ are i.i.d., $H(X_1, X_2, \ldots, X_n) = nH(X)$. □

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

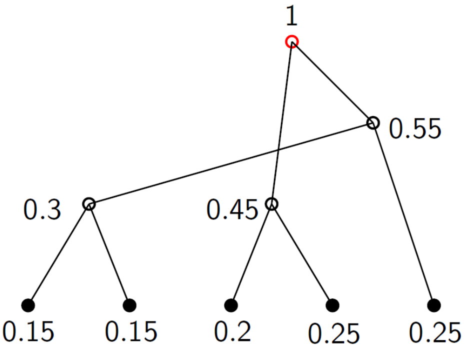
Step 1. Merge the D symbols with the smallest probabilities, and generate one new symbol whose probability is the summation of the D smallest probabilities.

Step 2. 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

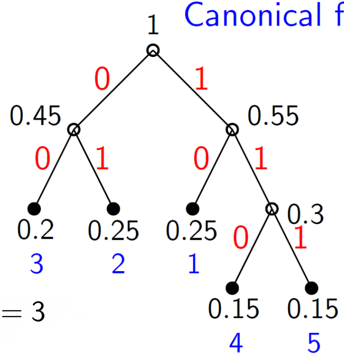
x	p(x)
1	0.25
2	0.25
3	0.2
4	0.15
5	0.15



Reconstruct the tree

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:

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

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

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

$$L \geq H_2(X)$$

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

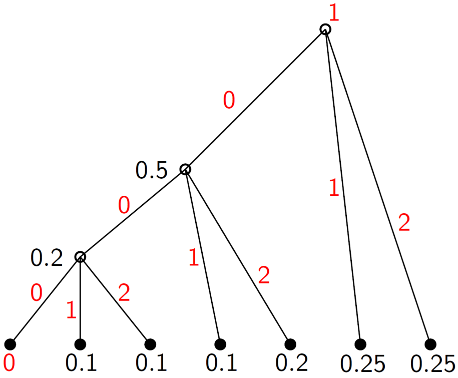
$\mathcal{D} = \{0, 1, 2\}$

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.

Huffman Codes: A few examples

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

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

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

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

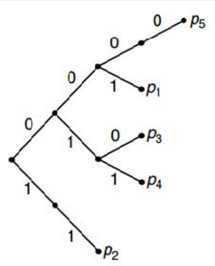
Thus, we must have $\ell_k \geq \ell_j$. □

Optimality of Huffman Codes

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

Proof.

If the two longest codewords are NOT of the same length, one can delete the last bit of the longer one, preserving the prefix property and achieving lower expected codeword length, contradiction! By property 1, the longest codewords must belong to the least probable source symbols.



- 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

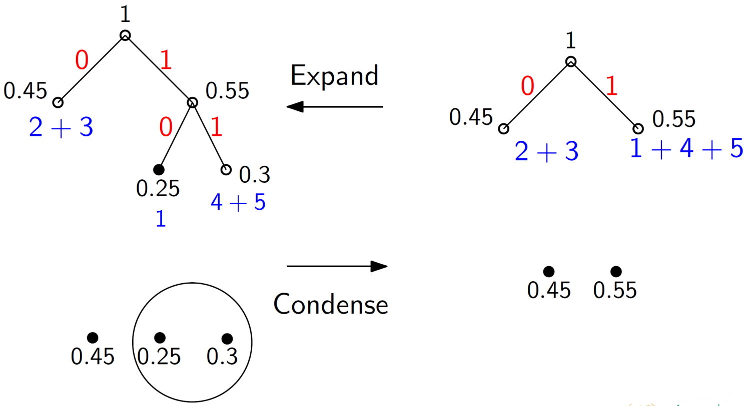
For any distribution, the optimal prefix codes (with minimum expected length) should satisfy the following properties:

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

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

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

Reading & Homework

Reading : 5.3 - 5.7

Homework : Problems 5.4, 5.6