INFORMATION THEORY & CODING

Week 5 : Source Coding 1

Dr. Rui Wang

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

Email: wang.r@sustech.edu.cn

October 17, 2023

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Review Summary

Theorem (AEP)

"Almost all events are almost equally surprising." Specifically, if $X_1, X_2, \ldots$ are i.i.d. $\sim p(x)$, then

$$-\frac{1}{n}\log p(X_1, X_2, \ldots, X_n) \to H(X) \text{in probability}.$$

Definition

The *typical set* $A_\epsilon^{(n)}$ is the set of sequences $x_1, x_2, \ldots, x_n$ satisfying

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)}.$$

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Review Summary

Properties of the typical set

1. If $(x_1, x_2, \ldots, x_n) \in A_\epsilon^{(n)}$, then $H(X) - \epsilon \leq -\frac{1}{n} \log p(x_1, x_2, \ldots, x_n) \leq H(X) + \epsilon$.
2. $\Pr[A_\epsilon^{(n)}] > 1 - \epsilon$ for n sufficiently large.
3. $|A_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$, where $|A|$ denotes the cardinality of the set A.
4. $|A_\epsilon^{(n)}| \geq (1-\epsilon)2^{n(H(X)-\epsilon)}$ for n sufficiently large.

Theorem

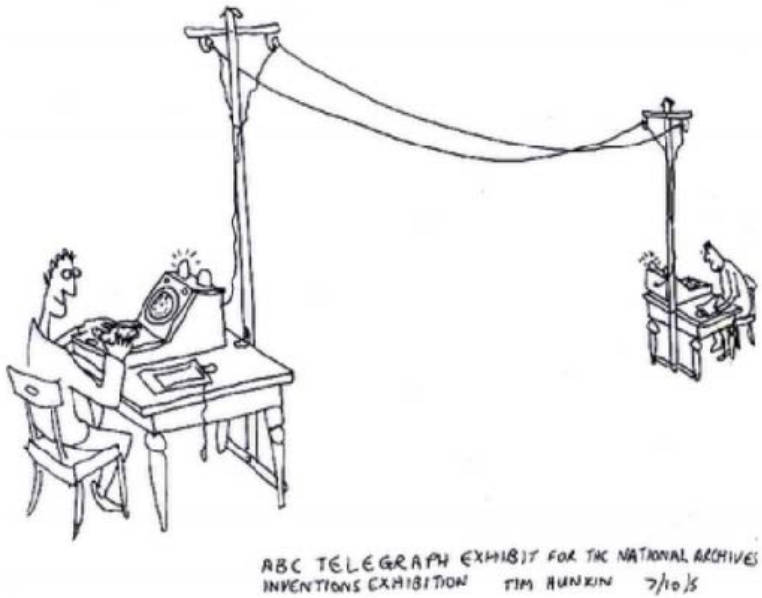
Let X^n be i.i.d. $\sim p(x)$. There exists a code that one-to-one maps sequences x^n of length n into binary strings with

$$E[\frac{1}{n}\ell(X^n)] \leq H(X) + \epsilon$$

for n sufficiently large.

Source Coding

Which horse won in the horse racing?



Source Coding

Which horse won in the horse racing?

X	Pr	Code I	Code II
0	1/2	000	0
1	1/4	001	10
2	1/8	010	110
3	1/16	011	1110
4	1/64	100	111100
5	1/64	101	111101
6	1/64	110	111110
7	1/64	111	111111

$$H(X) = -\sum p_i \log p_i = 2\text{bits}$$

Which code is better?

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Source Coding (Data Compression)

- We interpret that $H(X)$ is the best achievable data compression.

- We want to develop practical lossless coding algorithms that approach, or achieve the entropy limit $H(X)$.

Terminology

X	Pr	Code I	Code II
0	$1/2$	000	0
1	$1/4$	001	10
2	$1/8$	010	110
3	$1/16$	011	1110
4	$1/64$	100	111100
5	$1/64$	101	111101
6	$1/64$	110	111110
7	$1/64$	111	111111

- Source alphabet $\mathcal{X} = \{0, 1, 2, 3, 4, 5, 6, 7\}$.
- Code alphabet $\mathcal{D} = \{0, 1\}$.
- Codeword, e.g., 010 for $X = 2$ in Code 1.
- Codeword length, e.g., codeword length for Code 1 is 3.
- Codebook: all the codewords.

Source Coding

Notation (Alphabet Extension)

The set of all possible sequences based on a finite alphabet $\mathcal{D}$ is denoted by $\mathcal{D}^*$. E.g.,
$\mathcal{D} = \{0, 1\} \rightarrowtail \mathcal{D}^* = \{0, 1, 00, 01, 10, 11, 000, ...\}$.

Definition (Source Code)

Let $\mathcal{X}$ be the alphabet of a random variable X, and $\mathcal{D}$ be the alphabet of code. A *source code* C for the random variable X is a map

$$\begin{aligned} C: \quad & \mathcal{X} \to \mathcal{D}^* \\ & x \mapsto C(x) \end{aligned}$$

where $C(x)$ is the codeword associated with x. Let $\ell(x)$ denote the length of C(x).

Source Coding

Definition

The *expected length* $L(X)$ of a source code C for a random variable X with probability mass function $p(x)$ is

$$L(X) = E\ell(X) = \sum_{x \in \mathcal{X}} p(x)\ell(x).$$

X	Pr	Code I	Code II
0	1/2	000	0
1	1/4	001	10
2	1/8	010	110
3	1/16	011	1110
4	1/64	100	111100
5	1/64	101	111101
6	1/64	110	111110
7	1/64	111	111111

$$L_1(X) = 3$$
$$L_2(X) = 2$$

Source Coding Applications

- Magnetic recording: cassette, hard drive ...
- Speech compression
- Compact disk (CD)
- Image compression: JPEG

Source Coding: Set of codes

For $\mathcal{X} = \{1, 2, 3, 4\}$ and $\mathcal{D} = \{0, 1\}$, consider

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

- Code efficiency $= H(X)/E[\ell(X)]$
- Which code is best? Would we prefer C_I or C_{II}?

  Consider C_I and decode string: 00001. It would come from $1, 2, 1, 2, 3$ or $2, 1, 2, 1, 3$ or $1, 1, 1, 1, 3$, or etc.

Source Coding: Set of codes

For $\mathcal{X} = \{1, 2, 3, 4\}$ and $\mathcal{D} = \{0, 1\}$, consider

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

- Code efficiency $= H(X)/E[\ell(X)]$
- Which code is best? Would we prefer C_I or C_{II}?

  Consider C_{II} and decode string: 0011. It could be either $1, 1, 2, 2$ or $3, 4$.

Source Coding: Set of codes

For $\mathcal{X} = \{1, 2, 3, 4\}$ and $\mathcal{D} = \{0, 1\}$, consider

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

- Consider C_{III}. Can we decode 1100000000?

  Yes. But if we only see a prefix, such as 11, we don't know until we see more bits to the end.

  $1100000000 = 3, 2, 2, 2, 2$

  $11000000000 = 4, 2, 2, 2, 2$

Source Coding: Set of codes

For $\mathcal{X} = \{1, 2, 3, 4\}$ and $\mathcal{D} = \{0, 1\}$, consider

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

- Consider C_{IV}. This code seems at least feasible (since $E[\ell] \geq H$). Decoding seems easy: (e.g., $111110100 = 111, 110, 10, 0 = 4, 3, 2, 1$).

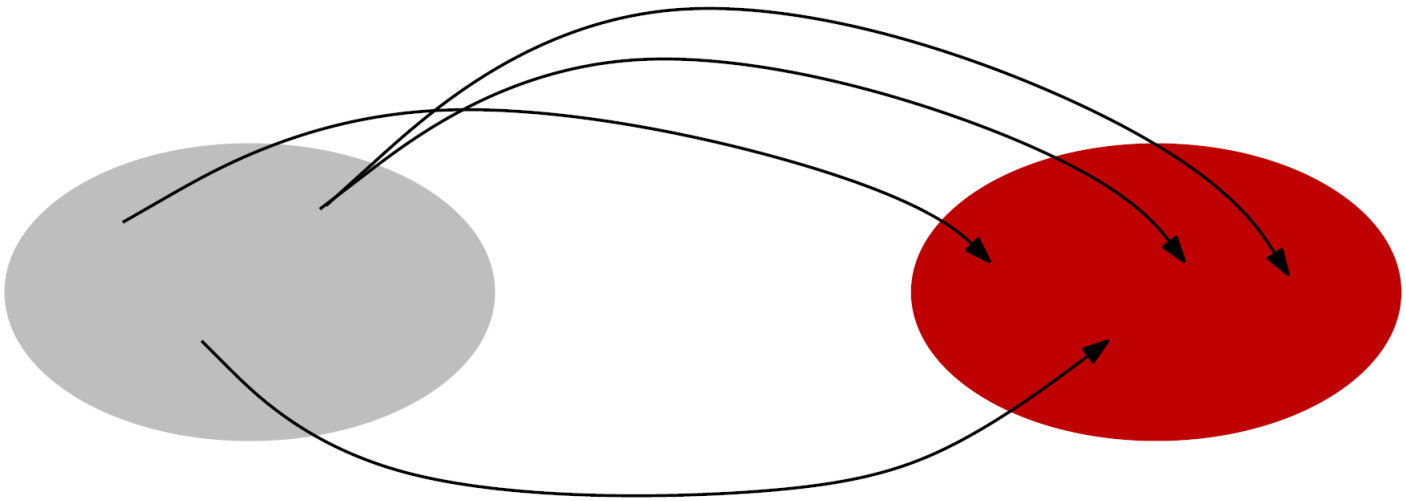
南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Source Coding: Code types

Definition (Nonsingular Code)

A code C is called *nonsingular* if every realization of $\mathcal{X}$ maps onto a difference codeword in $\mathcal{D}^*$, i.e.,

$$x \neq x' \Rightarrow C(x) \neq C(x').$$



Source Coding: Code types

Definition (Nonsingular Code)

A code C is called *nonsingular* if every element of $\mathcal{X}$ maps onto a difference string in $\mathcal{D}^*$, i.e.,

$$x \neq x' \Rightarrow C(x) \neq C(x').$$

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

C_I is singular.

Source Coding: Code types

Definition (Code Extension)

The *extension* of a code C: $\mathcal{X} \to \mathcal{D}^*$ is defined by

$$C(x_1 x_2 \cdots x_n) = C(x_1)C(x_2)\cdots C(x_n).$$

Definition (Unique Decodable Code)

A code is called *uniquely decodable* if its extension is nonsingular.

$$x_1 x_2 \ldots x_m \neq x_1' x_2' \ldots x_n' \Rightarrow C\left(x_1 x_2 \ldots x_m\right) \neq C\left(x_1' x_2' \ldots x_n'\right)$$

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Source Coding: Code types

Definition (Unique Decodable Code)

A code is called *uniquely decodable* if its extension is nonsingular.

C_{II}^* is singular. $(C(1,1) = C(3) = 00)$

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

C_I is singular.
C_{II} is NOT u.d..

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Source Coding: Code types

Definition (Unique Decodable Code)

A code is called *uniquely decodable* if its extension is nonsingular.

C_{III} is uniquely decodable.

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

C_I is singular.
C_{II} is NOT u.d..

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Source Coding: Code types

Definition (Unique Decodable Code)

A code is called *uniquely decodable* if its extension is nonsingular.

$$1100000000 = 3, 2, 2, 2, 2$$
$$11000000000 = 4, 2, 2, 2, 2$$

To know the source, we have to wait until the end!

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

C_I is singular.
C_{II} is NOT u.d..

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Source Coding: Code types

Definition (Prefix Code)

A code C is called a *prefix code* (a.k.a. *instantaneous*) iff no codeword of C is a prefix of any other codeword of C.

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

C_I is singular.
C_{II} is NOT u.d..
C_{III} is NOT prefix.
C_{IV} is prefix.

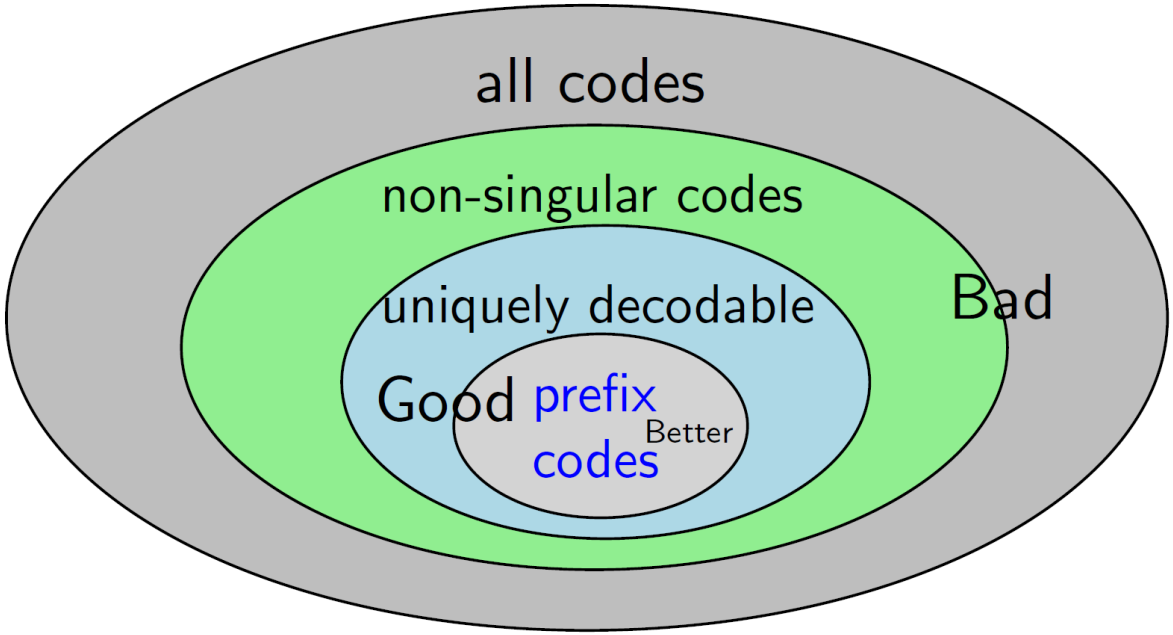
Source Coding: Code types

For $\mathcal{X} = \{1, 2, 3, 4\}$ and binary code, consider

x	$p(x)$	C_I	C_{II}	C_{III}	C_{IV}
1	1/2	0	0	10	0
2	1/4	0	1	00	10
3	1/8	1	00	11	110
4	1/8	10	11	110	111
$H(X)$	1.75	–	–	–	–
$E\ell(X)$	–	1.125	1.25	2.125	1.75

- C_I is singular.
- C_{II} is non-singular, but not uniquely decodable.
- C_{III} is non-singular, uniquely decodable, but NOT prefix.
- C_{IV} is non-singular, uniquely decodable, and prefix.

Source Coding: Classes of codes



- Goal: to find a prefix code with minimum expected length.

Kraft Inequality

Theorem 5.2.1 (Kraft Inequality)

For any prefix code over an alphabet of size D, the codeword lengths $\ell_1, \ell_2, \ldots, \ell_m$ must satisfy the inequality

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

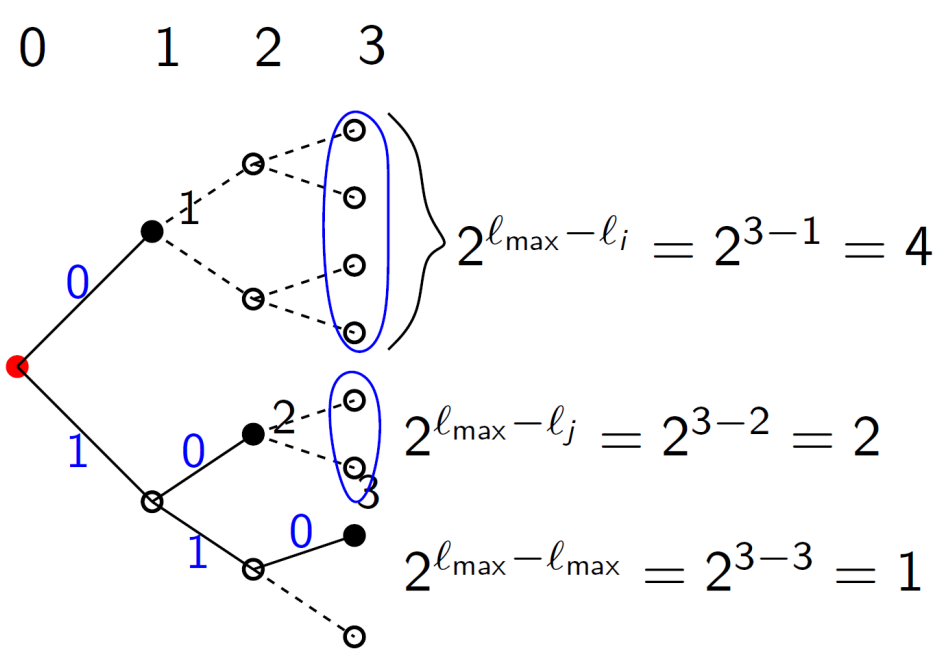
Conversely, given a set of codeword lengths that satisfy this inequality, there exists a prefix code with these codeword lengths.

Kraft Inequality

Proof Idea. (A small example) To prove: A prefix code with lengths $\ell_1, \ell_2, \ldots, \ell_m$, the inequality

$$\sum_i D^{-\ell_i} \leq 1 \qquad \text{holds.}$$

x	$c(x)$
1	0
2	10
3	110



$\ell_{\max} = 3$

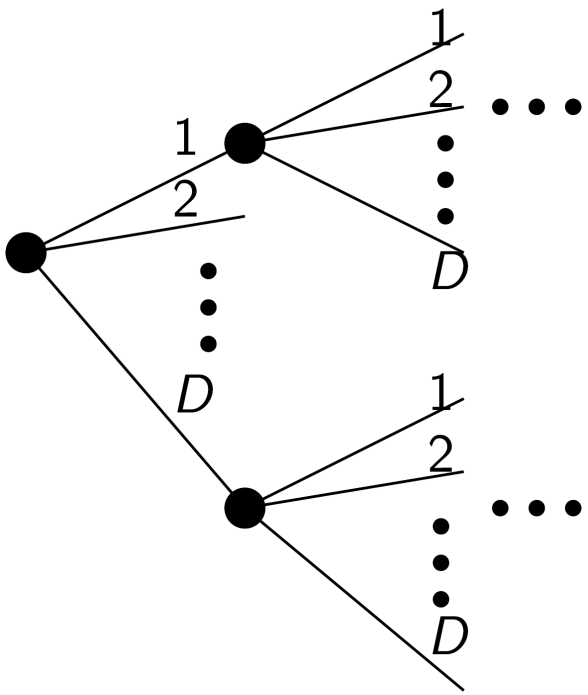
$$\sum_i 2^{-\ell_i} \leq 1 \Leftarrow \sum_i 2^{\ell_{\max}-\ell_i} \leq 2^{\ell_{\max}}$$

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Kraft Inequality

Proof. (in general)

- Represent the set of prefix codes on a D-ary tree:

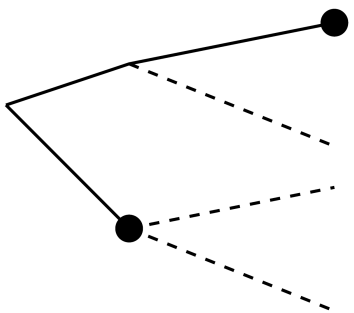


- Codewords correspond to leaves
- Path from root to each leaf determines a codeword
- Prefix condition: won't get to a codeword until we get to a leaf (no descendants of codewords are codewords)

Kraft Inequality

Proof. (in general)

- $\ell_{\max} = \max_i(\ell_i)$ is the length of the longest codeword.
- We can expand the full-tree down to depth $\ell_{\max}$:



The nodes at the level $\ell_{\max}$ are either

1. codewords
2. descendants of codewords
3. neither

- Consider a codeword i at depth ℓ_i in tree
- There are $D^{\ell_{\max}-\ell_i}$ descendants in the tree at depth $\ell_{\max}$
- Descendants of code i are disjoint from decedents of code j (prefix free condition)

Kraft Inequality

Proof. (in general)

- All the above implies:

$$\sum_i D^{\ell_{\max}-\ell_i} \leq D^{\ell_{\max}} \quad \Rightarrow \sum_i D^{-\ell_i} \leq 1$$

- Conversely: given codewords lengths $\ell_1, \ell_2, \ldots, \ell_m$ satisfying Kraft inequality, try to construct a prefix code.

Kraft Inequality

Proof. (in general)

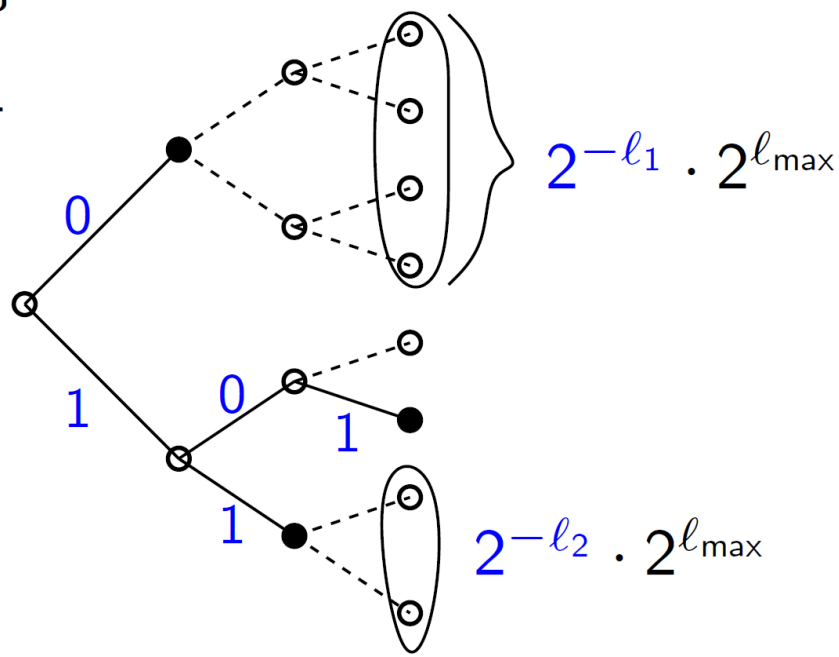
- Conversely: given codewords lengths $\ell_1, \ell_2, \ldots, \ell_m$ satisfying Kraft inequality, try to construct a prefix code.

$$\{\ell_1, \ell_2, \ell_3\} = \{1, 2, 3\}$$

$$2^{-1} + 2^{-2} + 2^{-3} \leq 1$$

x	$c(x)$
1	0
2	11
3	101

C is prefix.



Kraft Inequality

Proof. (in general)

- Conversely: given codewords lengths $\ell_1, \ell_2, \ldots, \ell_m$ satisfying Kraft inequality, try to construct a prefix code.

Left as an Exercise.

Reading & Homework

Reading : 5.1, 5.2

Homework : Problems 5.1, 5.3

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 23, 2023

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

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

- **Extended Kraft inequality for prefix code**
- **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.

Extended Kraft Inequality

Theorem 5.5.1 (Extended Kraft Inequality)

Kraft inequality holds also for all countably infinite set of codewords, i.e., the codeword lengths satisfy the extended Kraft inequality,

$$\sum_{i=1}^{\infty} D^{-\ell_i} \leq 1$$

Conversely, given any $\ell_1, \ell_2, \ldots$ satisfying the extended Kraft inequality, we can construct a prefix code with these codeword lengths.

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Extended Kraft Inequality

Theorem 5.2.2 (Extended Kraft Inequality)

Kraft inequality holds also for all countably infinite set of codewords.

Proof.

Consider the ith codeword $y_1 y_2 \cdots y_{\ell_i}$. Let $0.y_1 y_2 \cdots y_{\ell_i}$ be the real number given by the D-ary expansion

$$0.y_1 y_2 \cdots y_{\ell_i} = \sum_{j=1}^{\ell_i} y_j D^{-j},$$

which corresponds to the interval

$$[0.y_1 y_2 \cdots y_{\ell_i}, 0.y_1 y_2 \cdots y_{\ell_i} + \frac{1}{D^{\ell_i}}).$$

□

Extended Kraft Inequality

Theorem 5.2.2 (Extended Kraft Inequality)

Kraft inequality holds also for all countably infinite set of codewords.

Proof. (cont.)

By the prefix condition, these intervals are disjoint in the unit interval $[0, 1]$. Thus, the sum of their lengths is ≤ 1. This proves that

$$\sum_{i=1}^{\infty} D^{-\ell_i} \leq 1.$$

For converse, reorder indices in increasing order and assign intervals as we walk along the unit interval. □

Kraft Inequaltiy for Uniquely Decodable Codes

Theorem 5.2.3 (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

$$\begin{aligned}\left(\sum_{x\in\mathcal{X}} D^{-\ell(x)}\right)^k &= \sum_{x_1\in\mathcal{X}}\sum_{x_2\in\mathcal{X}}\cdots\sum_{x_k\in\mathcal{X}} D^{-\ell(x_1)}D^{-\ell(x_2)}\cdots D^{-\ell(x_k)} \\ &= \sum_{x_1,x_2,\cdots x_k\in\mathcal{X}^k} D^{-\ell(x_1)}D^{-\ell(x_2)}\cdots D^{-\ell(x_k)} \\ &= \sum_{x^k\in\mathcal{X}^k} D^{-\ell(x^k)}\end{aligned}$$

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.)

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$\to 1$ as $k \to \infty$, 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*

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

Proof.

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

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

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

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

□

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., $H(X_1, X_2, \ldots, X_n) = nH(X)$. □

Textbook

Related Sections : 5.3 - 5.5

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

October 31, 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

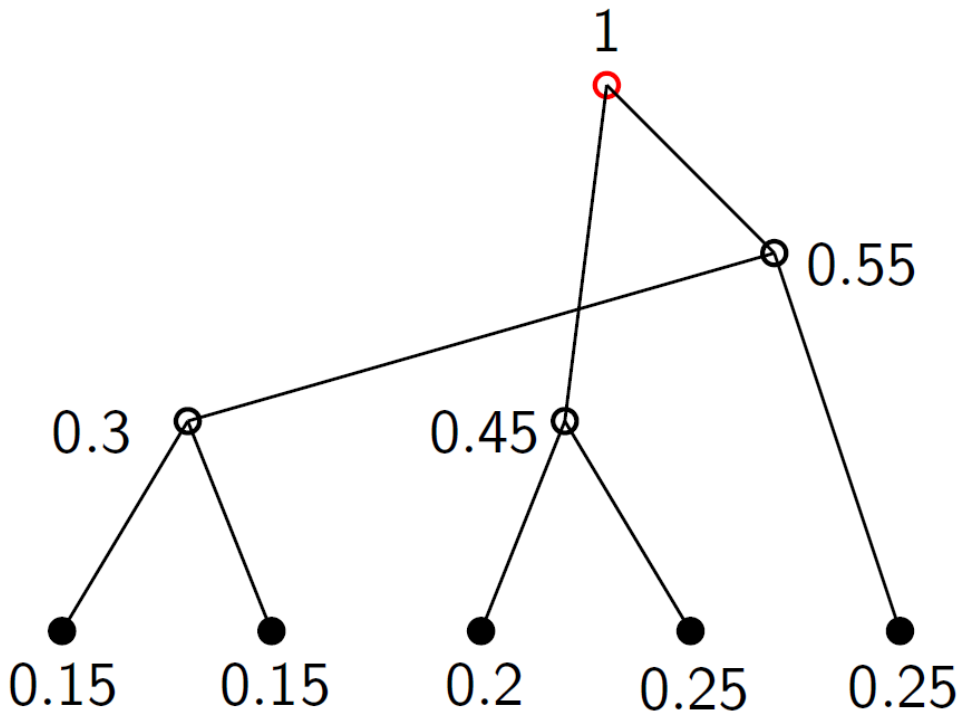
- Merge the D symbols with the smallest probabilities, and generate one new symbol whose probability is the summation of the D smallest probabilities.
- Assign the D corresponding symbols with digits $0, 1, \ldots, D-1$, then go back to Step 1.

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

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

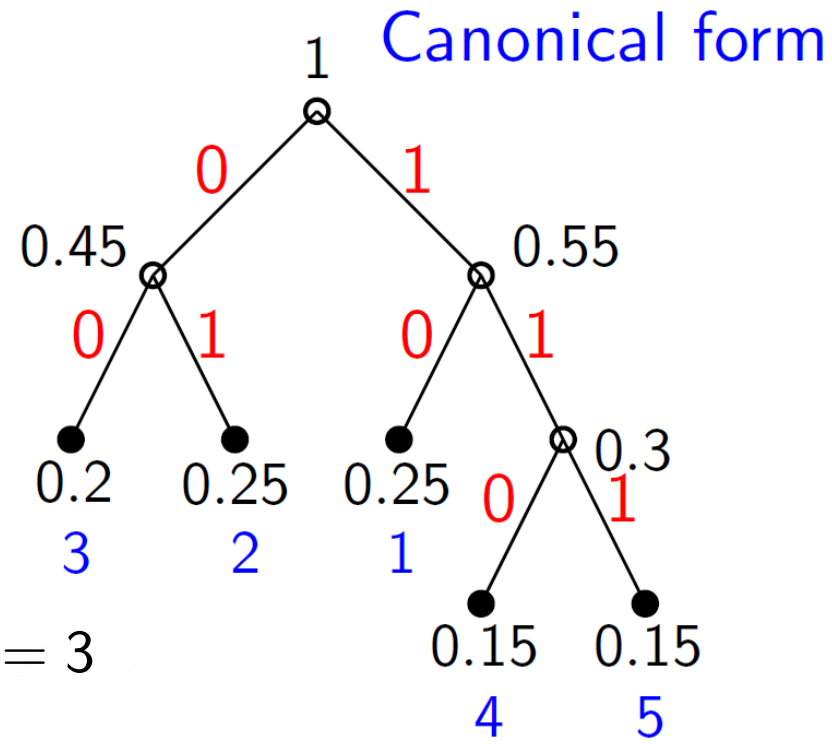


Reconstruct the tree

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



Validations:

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

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

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

$$L \geq H_2(X)$$

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

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

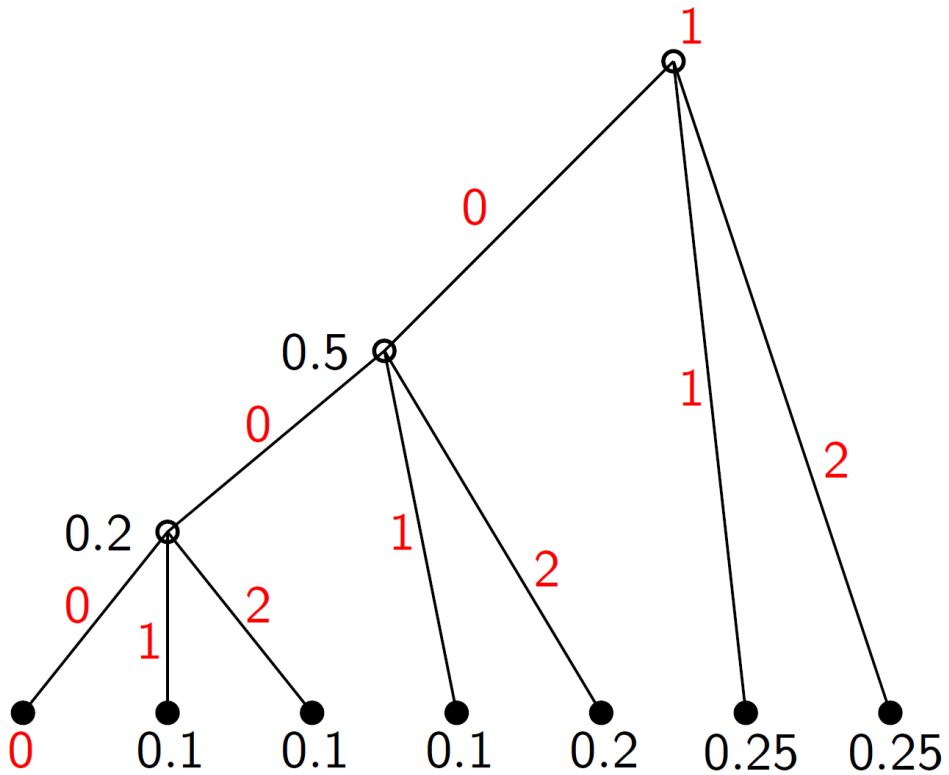
$\mathcal{D} = \{0, 1, 2\}$

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.

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.*

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

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

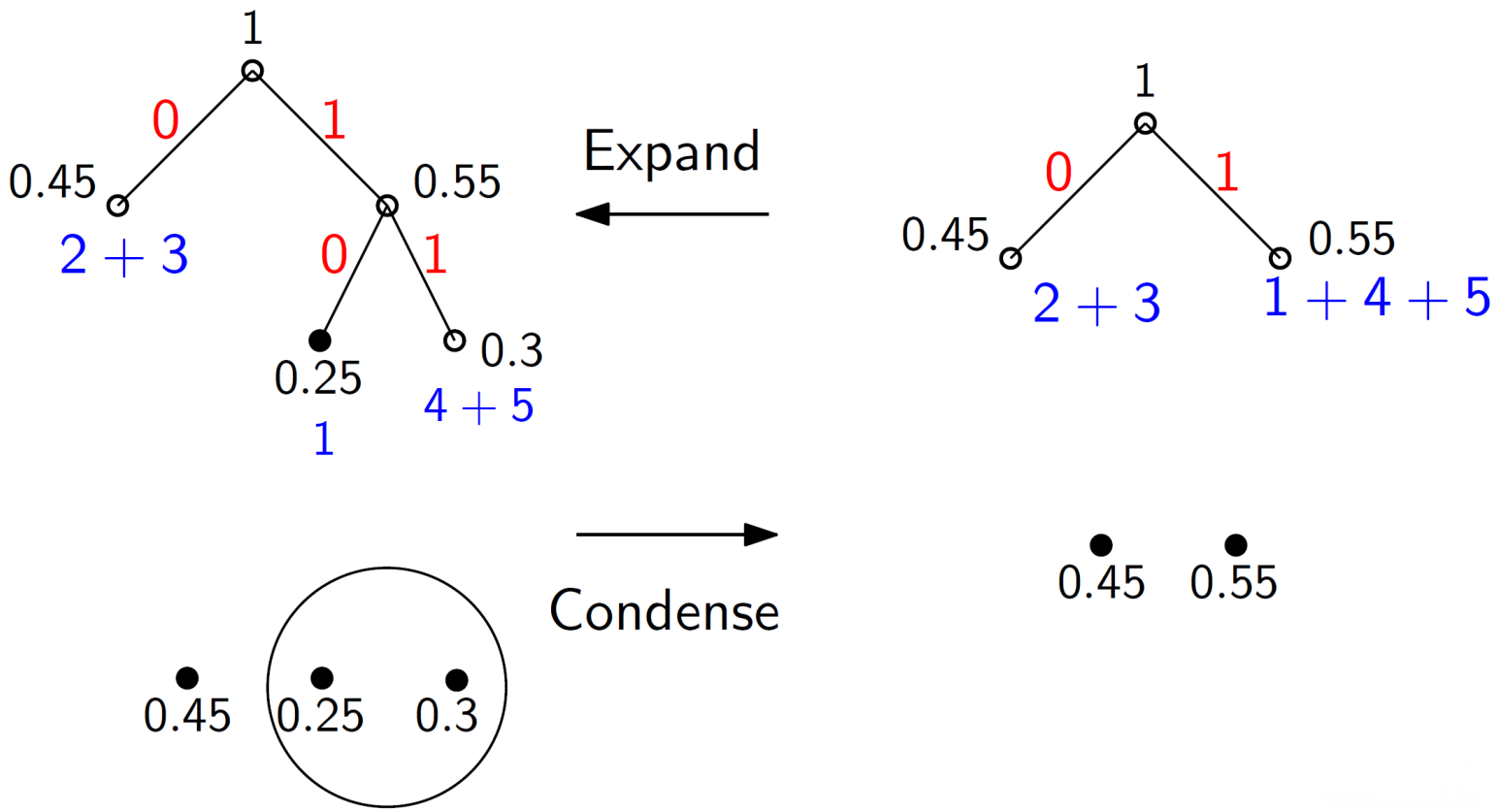
For any distribution, the optimal prefix codes (with minimum expected length) should satisfy the following properties:

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

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

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

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Textbook

Related Sections : 5.6 - 5.8

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

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

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

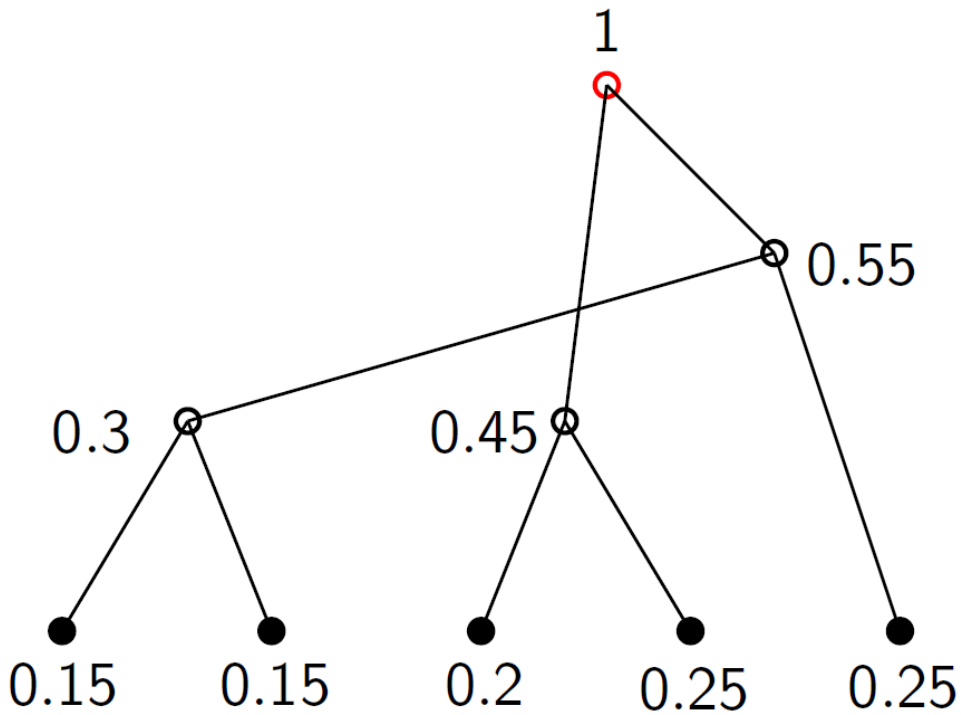
- Merge the D symbols with the smallest probabilities, and generate one new symbol whose probability is the summation of the D smallest probabilities.
- Assign the D corresponding symbols with digits $0, 1, \ldots, D-1$, then go back to Step 1.

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

Huffman Codes: A few examples

Example 1

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

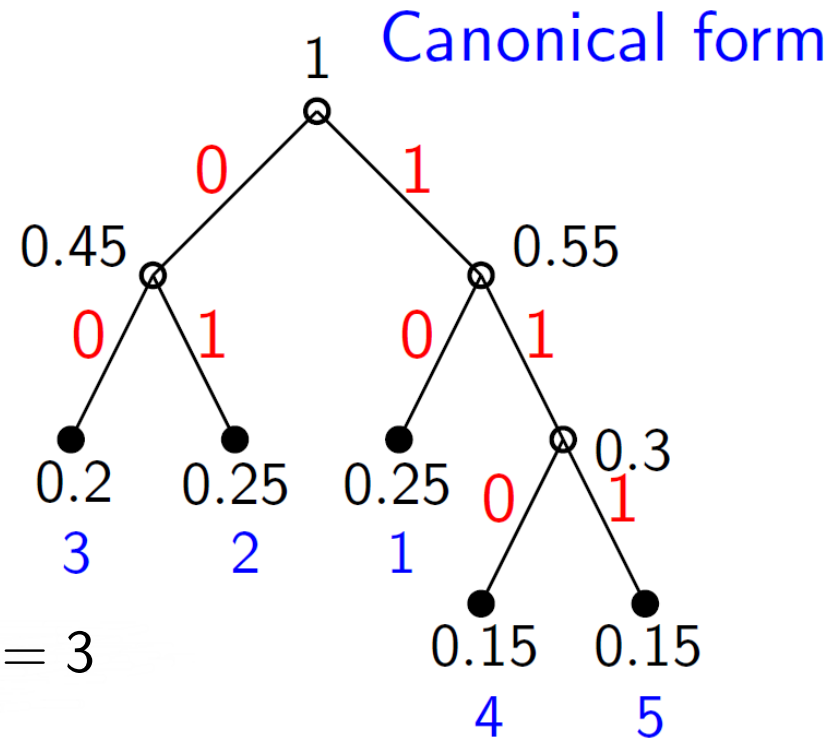


Reconstruct the tree

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



Validations:

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

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

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

$$L \geq H_2(X)$$

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

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

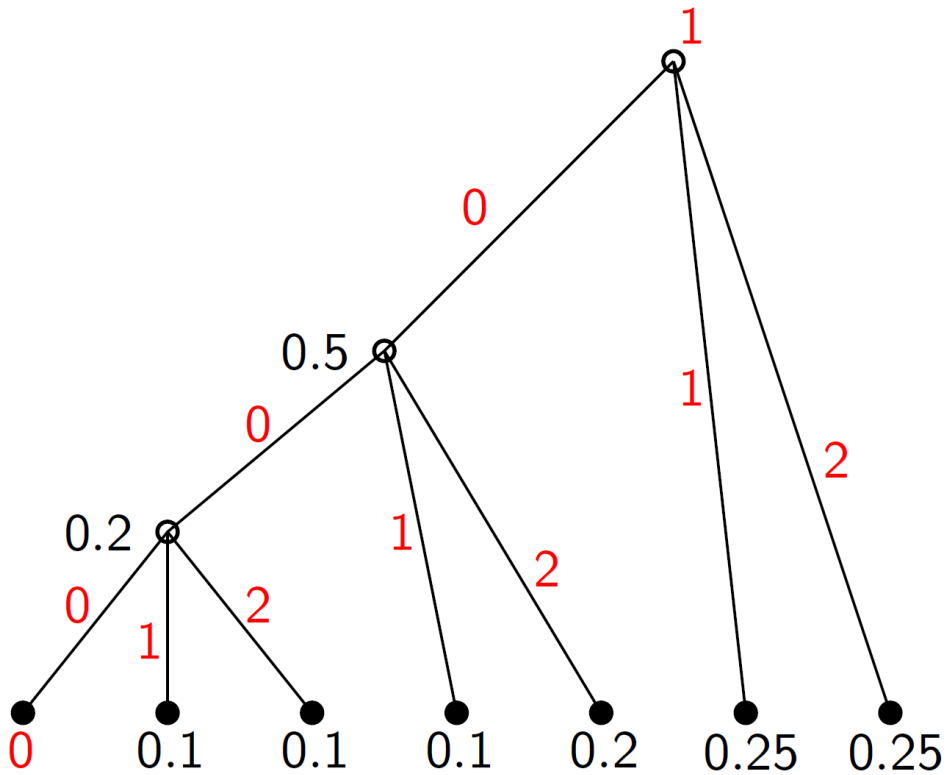
$\mathcal{D} = \{0, 1, 2\}$

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.

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.*

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

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

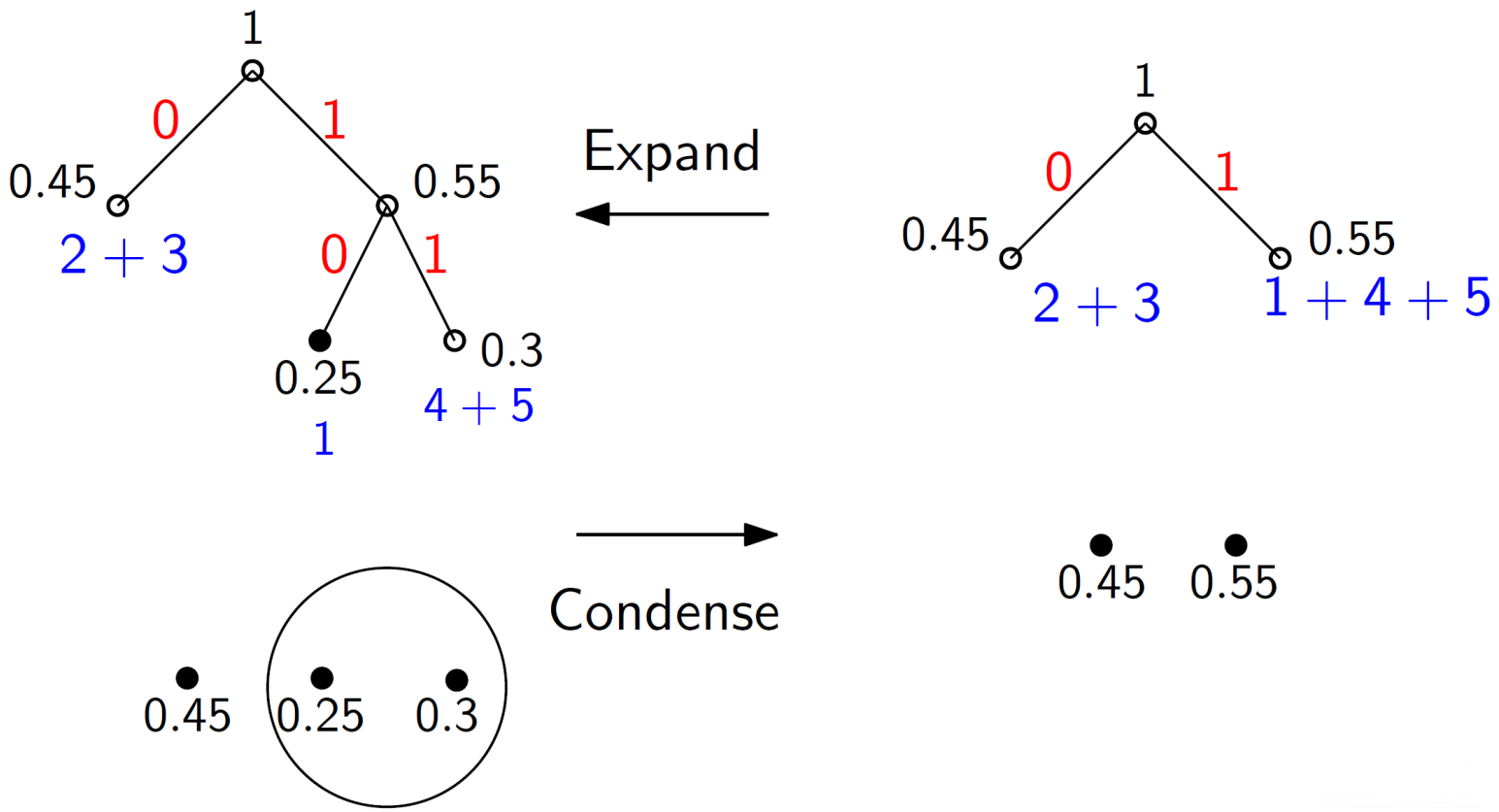
For any distribution, the optimal prefix codes (with minimum expected length) should satisfy the following properties:

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

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

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

南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Textbook

Related Sections : 5.6 - 5.8