

## 5. Data Compression

### \* Data Compression

- Entropy : 암축 한계.
- 데이터 암축 code : 차례 나오는 sequence에 같은 codeword. 드물게 : 긴 codeword 허용.
  - sequence : 발생 data. codeword : 암축 결과.

### \* Source Code

- RV  $X$ 에 대한 source code  $C$  :  $X \rightarrow D^*$   $\cong$  mapping
  - $C(X) = x \in X$ 에 대한 codeword.  $C(x) \in D^*$
  - $l(x) = C(x)$ 의 길이.
  - Expected length
    - $C(x)$ 에 대한  $L(C)$

$$L(C) = \sum_{x \in X} p(x)l(x) = E[l(x)]$$

- $E[l(x)]$  or  $H(X)$  간 비교하기 :  $H(X)$ 는 probability로만 계산됨 (codeword 상관X)
  - $E[l(x)]$ 는 실제 codeword의 길이로 계산됨.
  - $E[l(x)]$ 가 가장 짧은 code = optimal code.

### \* Non singularity $\leftrightarrow$ Singular : 구분이 불가능함.

- $X$ 의 element들에 대해 다음을 만족하는 code는 nonsingular.
  - $x \neq x' \Rightarrow C(x) \neq C(x')$

### \* Self punctuating code

- code 간의 "," 가 필요 없는 것.
- instantaneous code.

## \* Code extension

- $C(x_1 x_2 \dots x_n) = C(x_1) C(x_2) \dots C(x_n)$  : concatenation 가능.

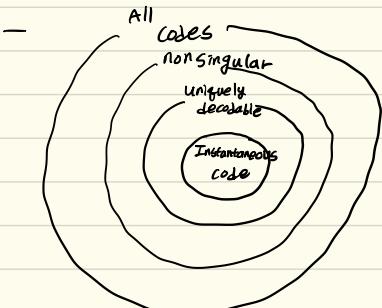
## \* Uniquely decodable codes

- 연결된 codeword에 대해, 유일한 조합으로 분해됨.

## \* Prefix code (= Instantaneous code or self punctuating code)

- 어떠한 codewords도 다른 codewords의 일부분이 아님.
- Codeword의 끝이 바로 보이는 것.
  - Uniquely decodable : 진행할 때는 여러 후보가 생길 수 있음, but 끝까지 봤을 때는 하나.

## \* 전처리



	x	Sing	Ninsing	Uniquely	Instantaneous
1	0	0	10	0	
2	0	010	00	10	
3	0	01	11	110	
4	0	10	110	111	

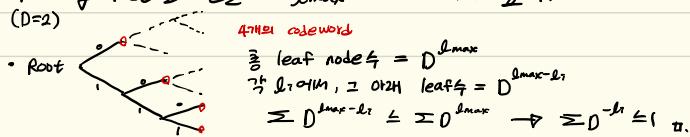
ex) 1100은 uniquely로 32인데,  
1101121 봤을 때 딱히 포함 (혹보: 4 or 32)

## \* Kraft inequality

- Prefix code (D)에 대해 codeword length가  $l_1, l_2, \dots$  면

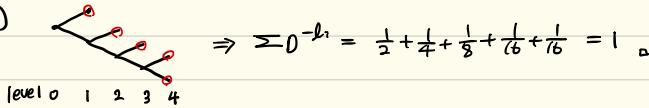
$$\sum_i D^{-l_i} \leq 1$$

Pf) D-ary tree를 만들자.  $l_{\max}$ 는 tree 최대높이.  
( $D=2$ )



- Converse ; equality를 만족하는 ( $\sum D^{-l_i} = 1$ ) code를 생성할 수 있음.

pf)



level 0 1 2 3 4

- Extended Kraft inequality

- $l_{\max}$ , codeword 개수를 infinite?

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

pf) 구간을 할당한다. real number in D-adic expansion

$$0.y_1y_2\dots y_{l_i} = \sum_{j=1}^{l_i} y_j D^{-j}$$

$\hookrightarrow [0.y_1y_2\dots y_{l_i}, 0.y_1y_2\dots y_{l_i} + \frac{1}{D^{l_i}}]$  할당. size of interval =  $D^{-l_i}$

$$\text{ex) } 0.11 \ (D=2) \Rightarrow [0.11, 0.11 + \frac{1}{2^2}] = [0.11, 1] \text{ 할당. }$$

$$\text{모든 구간 합} = \sum_{i=1}^{\infty} D^{-l_i} \leq 1 \quad \leftarrow [0, 1]$$

겹치지 않음.

instantaneous property.

### \*Optimal codes

- 가장 짧은  $\in \mathcal{L}(x)$  를 갖는 prefix code 만들기.

$$\cdot L(C) = E[l_i] = \sum p_i l_i$$

• optimal code : 가장 짧은  $L(C)$ 를 갖는 instantaneous code.

:  $H(x)$  와 비슷한 optimal임.

- Minimize  $L = \sum p_i l_i$  subject to  $\sum D^{-l_i} \leq 1$  인  $l_i$  찾기

• Lagrange multipliers 사용

$$\cdot J = \sum p_i l_i + \lambda (\sum D^{-l_i})$$
에 대해

$$\frac{\partial J}{\partial l_i} = p_i - \lambda D^{-l_i} \log_e D = 0, \quad D^{-l_i} = \frac{p_i}{\lambda \log_e D}$$

$$\sum D^{-l_i} = \sum \frac{p_i}{\lambda \ln D} = \frac{1}{\lambda \ln D} = 1, \quad \lambda = \ln D$$

$$\Rightarrow p_i = D^{-l_i}$$

- optimal code length

$$\cdot l_i^* = -\log_2 p_i \quad \begin{matrix} \rightarrow & p_i \uparrow \downarrow ; l_i^* \uparrow \\ & \uparrow ; \downarrow \end{matrix}$$

$$\cdot L^* = \sum p_i l_i^* = -\sum p_i \log_2 p_i = H(X)$$

ex)  $p_i = \frac{1}{6}$  ?  $\Rightarrow l_i^* = -\log_2 \frac{1}{6} = 2.22 \rightarrow 3 \text{ bit } \underline{\text{문자}}$   
 $\Rightarrow l_i = \lceil -\log_2 p_i \rceil$  but not optimal.

\* Optimality

$$- L \geq H_0(X) , \text{ equal } p_i = D^{-l_i} \quad (l_i = -\log_2 p_i)$$

$$PF) L - H_0(X)$$

$$\begin{aligned} &= \sum p_i l_i - \sum p_i \log_2 \frac{1}{p_i} \\ &= -\sum p_i \log_2 D^{-l_i} + \sum p_i \log_2 p_i \\ D^{-l_i} &= \frac{D^{-l_i}}{C} \cdot C, \quad C = \sum D^{-l_i} \text{ et } \overline{\text{으로}} \end{aligned}$$

$$\begin{aligned} L - H_0(X) &= \sum p_i \log_2 \frac{p_i}{l_i} - \sum p_i \log_2 C \\ &= D(\overbrace{p_i l_i}^{\geq 0}) + \log_2 \frac{1}{C} \geq 0 \quad \square \\ &\text{equality : } p_i = l_i = \frac{D^{-l_i}}{C} \\ C = 1, \quad p_i &= D^{-l_i} \end{aligned}$$

\* Shannon code : Not optimal.

- Optimal length

$$\cdot l_i = -\log_2 p_i$$

$$\cdot \text{integer} : l_i = \lceil -\log_2 p_i \rceil : \text{optimal 은 아님.}$$

## \* Bounds on optimal length

-  $l_1^*, \dots, l_m^*$   $\rightarrow$  optimal length

$$\cdot H_0(X) \leq L^* < H_0(X) + 1$$

*Shannon 정.*

$$PF) l_i = \lceil \log_2 \frac{1}{p_i} \rceil \text{ 이면}$$

$$\log \frac{1}{p_i} \leq l_i < \log \frac{1}{p_i} + 1$$

$$\Rightarrow p_i \log \frac{1}{p_i} \leq p_i l_i < p_i (\log \frac{1}{p_i} + 1)$$

$$H_0(X) \leq L < H_0(X) + 1 \rightarrow H_0(X) \leq L^* < L < H_0(X) + 1$$

- 2nd symbol (bound improved: 증명)

•  $Y = X_1, X_2, \dots, X_n$  이면

$$I(X_1, X_2, \dots, X_n) = I(Y)$$

$$P(X_1, X_2, \dots, X_n) = \prod_i P(X_i)$$

$$\Rightarrow L_n = \frac{1}{n} \sum P(X_1, \dots, X_n) I(X_1, \dots, X_n) = \frac{1}{n} \underbrace{EI(X_1, \dots, X_n)}_L$$

$$\Rightarrow 위의 증명에 대비 H(Y) \leq L \leq H(Y) + 1$$

$$H(X) \leq L_n \leq H(X) + \frac{1}{n}$$

$\rightsquigarrow n \rightarrow \infty : H(X)$ 을 수렴

## \* 잘못된 code에 의한 압축

- source의 정확한 distribution을 모르거나, 디코더가 같은 경우 miss match이 있는 때

- source :  $P(x)$ , codeword length =  $\lceil \log_2 \frac{1}{g(m)} \rceil$  이면

$$\cdot H(p) + D(p||g) \leq E_p l(X) \leq H(p) + D(p||g) +$$

*penalty*

$$PF) EI(X) = \sum p(x) \lceil \log \frac{1}{g(m)} \rceil < \sum p(x) (\log \frac{1}{g(m)} + 1)$$

$$= \sum p(x) \log \frac{1}{g} + \sum p \log \frac{1}{p} + 1$$

$$= D(p||g) + H(p) + 1$$

### \* Kraft Inequality for Uniquely Decodable code

- 짧은 코드는 Instantaneous code > Uniquely decodable code 이므로

Uniquely의 성능이 더 좋다면, 이게 나옴.

- Uniquely decodable code  $\Sigma 0^{-l_i} \leq 1$

Kraft Inequality 만족.

- But, codeword length 측면에서 성능 gain이 있진 않음. Uniquely를 쓴다고 더 짧아지는게 아님.

### \* Huffman Code : prefix code + Optimal codeword length

ex) X prob

X	prob		length	
1	0.25	0.3	1	$\rightarrow 01$ $x0.25$
2	0.125	0.25	2	$\rightarrow 10$ $x0.125$
3	0.2	0.25	2	$\rightarrow 11$ $x0.12$
4	0.15	0.2	3	$\rightarrow 000$ $x0.15$
5	0.15		3	$\rightarrow 001$ $x0.15$
				= 2.3 bits

- Ternary code ( $D=3$ ) : 각 단계에서 3개를 뜯음. 0, 1, 2 할당.

: 단위: ternary digits

: (비교) 2.3 bits, 1.5 ternary digits

0.15

$\Rightarrow$  표현 가능 수 :  $2^{2.3} = 4.92, 3^{1.5} = 5.20$

- Ternary에서는 dummy symbol 을 넣어서 (혹은 0) 개수를 맞추기도 함.

•  $1 + (D-1)K$  개 맞추기.

• 마지막에 2개 남아서, 더하는게 비효율적임 (길이 측면에서)

• binary는 필요X.

- 1개만 나오는건 아님. 여러개 가능.

## \* Huffman code의 Optimality

- Shannon code 보다 같거나 더 좋다.

ex)  $p: (\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12})$

$$\begin{array}{ccccccc} \text{huffman : } & \frac{1}{3} & - & \frac{1}{3} & - & \frac{1}{3} & \xrightarrow{\cdot 1} \\ & \frac{1}{3} & - & \frac{1}{3} & - & \frac{2}{3} & \xrightarrow{\cdot 10} \\ & \frac{1}{4} & \xrightarrow{\cdot 1} & \frac{1}{3} & & & \xrightarrow{\cdot 110} \\ & \frac{1}{12} & & & & & \xrightarrow{\cdot 111} \end{array}$$

3번째는 shannon이 더 좋음.  
average는  $2 < 2.173$   
but Huffman 습.

$$\text{Shannon : } l_i = \lceil -\log p_i \rceil \quad : \quad (2, 2, 2, 4)$$

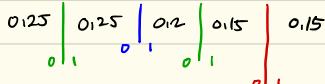
## \* Fano's construction

- Suboptimal procedure

ex)  $\{0.25, 0.25, 0.12, 0.15, 0.15\}$

1. 내림차순  $0.25 \ 0.25 \ 0.12 \ 0.15 \ 0.15$

2. 반복 차례는데, 두 덩어리 코데이트가 합쳐지는 무게 차례



3. 반복 ↗

4. Codeword : 00 01 10 110 111

- Optimal 아님.  $L(C) < H(X) + 2$

## \* Huffman code 특성

- Canonical code (정적 code)

• optimal instantaneous code 는 다음을 만족

① length는 확률에 반대.

② 가장 긴 codeword 개수는 같은 글자.

③ " 2개는 맨 마지막 bit만 다른.

- Huffman code는 optimal이다. ( $C^*$ )

• Huffman은 Holst 가 아님!

• 어떤 uniquely decodable  $C'$ 이 대해서도  $L(C^*) \leq L(C')$ .