CS258: Information Theory

Fan Cheng
Shanghai Jiao Tong University

http://www.cs.sjtu.edu.cn/~chengfan/
chengfan@sjtu.edu.cn
Spring, 2020

Outline

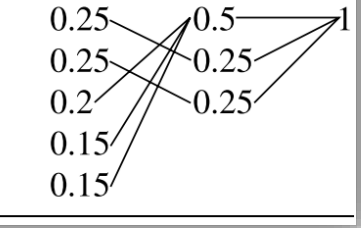
- Kraft inequality
- Optimal codes
- Huffman coding
- Shannon-Fano-Elias coding
- Generation of discrete distribution
- Universal source coding

Huffman Codes: Algorithm

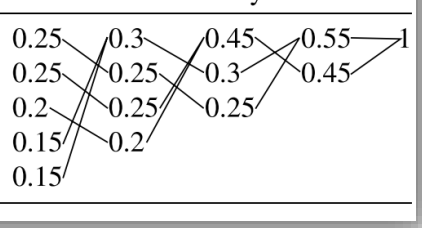
***D*-ary Huffman codes (prefix code)** for a given distribution:

- Each time combine D symbols with the lowest probabilities into a single source symbol, until there is only one symbol

Codeword	X	Probability
1	1	0.25
2	2	0.25
00	3	0.2
01	4	0.15
02	5	0.15



Codeword Length	Codeword	X	Probability
2	01	1	0.25
2	10	2	0.25
2	11	3	0.2
3	000	4	0.15
3	001	5	0.15



Huffman coding is optimal:

$$\min \sum p_i l_i$$

- Huffman coding for weighted codewords w_i:

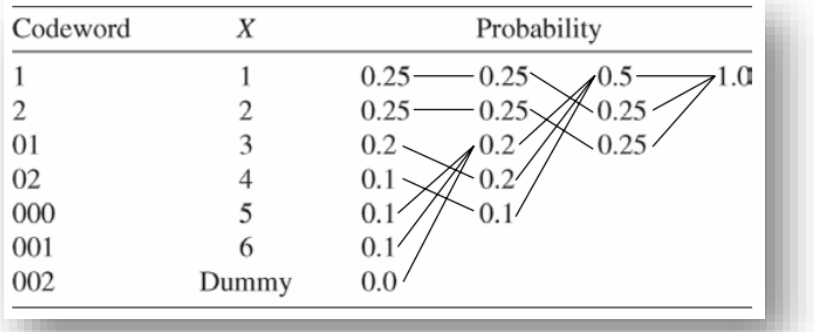
$$p_i \Rightarrow w_i \rightarrow \frac{w_i}{\sum w_i}$$

Huffman's algorithm for minimizing $\sum w_i l_i$ can be applied to any set of numbers $w_i \geq 0$

Huffman Codes: Algorithm

If $D \geq 3$, we may not have a sufficient number of symbols so that we can combine them D at a time. In such a case, we **add dummy symbols to the end of the set of symbols**. The dummy symbols have probability 0 and are inserted to fill the tree.

- Since at each stage of the reduction, the number of symbols is reduced by $D-1$, we want the total number of symbols to be $\mathbf{1 + k(D - 1)}$, where k is the number of merges.



- Morse Vs. Huffman

  Morse code could be regarded as a certain Huffman code when $p_i's$ are estimated

- Adaptive Huffman coding

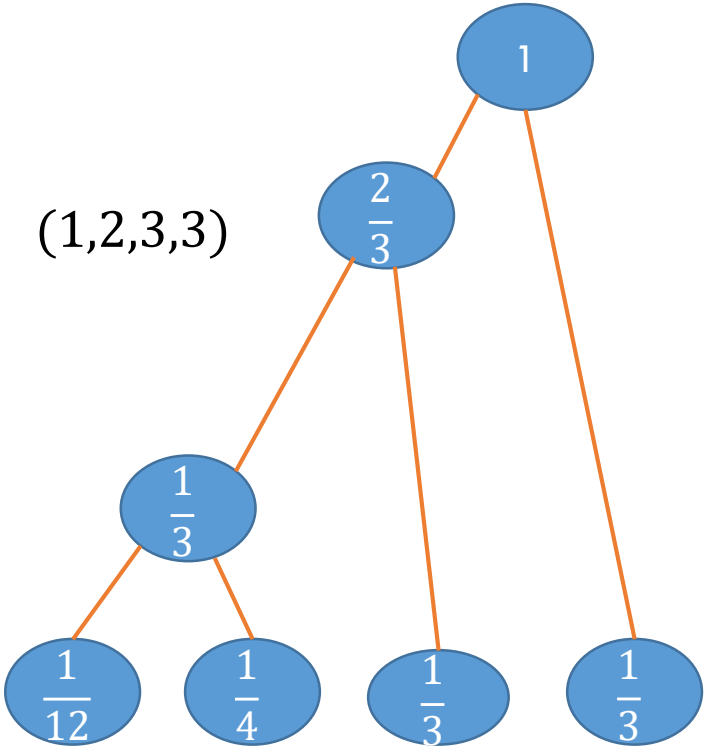
Huffman Codes: Extension

Huffman code is not unique: $l_i, 1 \le i \le n$

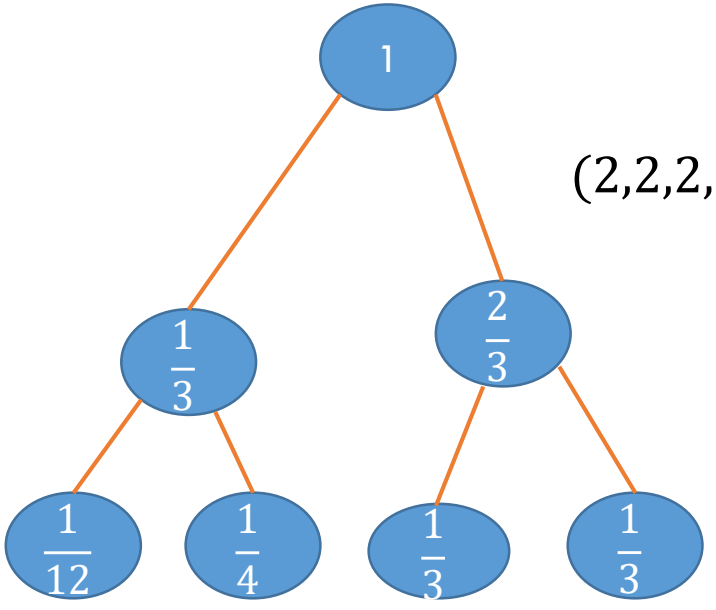
- Counterexample: $0 \to 1, 1 \to 0$
- For $p(X) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{12}\right)$, both $(2,2,2,2)$ and $(1,2,3,3)$ are optimal Huffman code

Number of different Huffman trees?

$(1,2,3,3)$



$(2,2,2,2)$



Huffman Codes: Extension

A probability distribution $\Pr(X)$ is called ***D-adic*** if each of the probabilities

$$\Pr(X = x_i) = D^{-n}$$

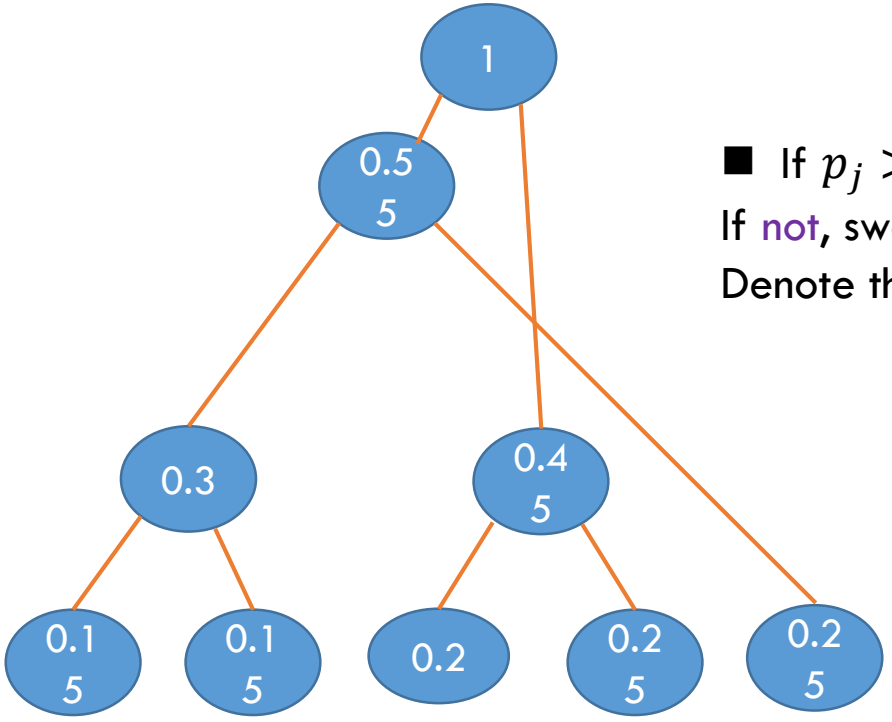
for some n.

- For a $D\text{-}adic$ distribution, the optimal solution in Lagrange is unique: $l_i = \log\frac{1}{p_i} = n_i$

- Huffman Vs. Shannon codes
  - Shannon codes $\left\lceil \log\frac{1}{p_i} \right\rceil$ attain optimality within 1 bit. If the prob. distribution is ***D-adic***, Shannon codes are optimal
  - Shannon codes may be much worse when $p_i \to 0$: Consider two symbols, one with probability 0.9999 and the other with probability 0.0001. The optimal codeword length is 1 bit for both symbols. The lengths of Shannon codes are 1 and 14.
- Huffman codes in application
  - JPEG, PNG, ZIP, MP3
  - Cryptography
  - Internet protocol, HTTP header (RFC)

Canonical Codes: I

Without loss of generality, we will assume that the probability masses are ordered, so that $p_1 \geq p_2 \geq \cdots \geq p_m$. Recall that **a code is optimal if $\sum p_i l_i$ is minimal**. For any optimal coding scheme



- If $p_j > p_k$, then $l_j \leq l_k$.

If not, swap the codewords of j and k.

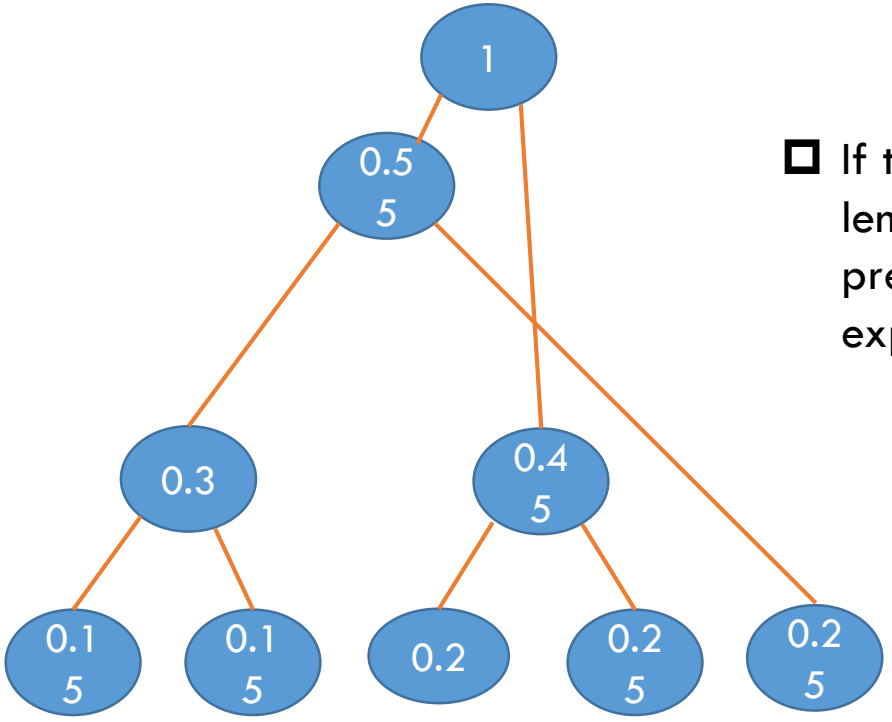
Denote the new code by C'_m

$$\begin{aligned} L(C'_m) - L(C_m) &= \sum p_i l'_i - \sum p_i l_i \\ &= p_j l_k + p_k l_j - p_j l_j - p_k l_k \\ &= (p_j - p_k)(l_k - l_j) < 0 \end{aligned}$$

1. The lengths are ordered inversely with the probabilities (i.e., if $p_j > p_k$, then $l_j \leq l_k$).

Canonical Codes: II

Without loss of generality, we will assume that the probability masses are ordered, so that $p_1 \geq p_2 \geq \cdots \geq p_m$. Recall that **a code is optimal if $\sum p_i l_i$ is minimal**. For any optimal coding scheme

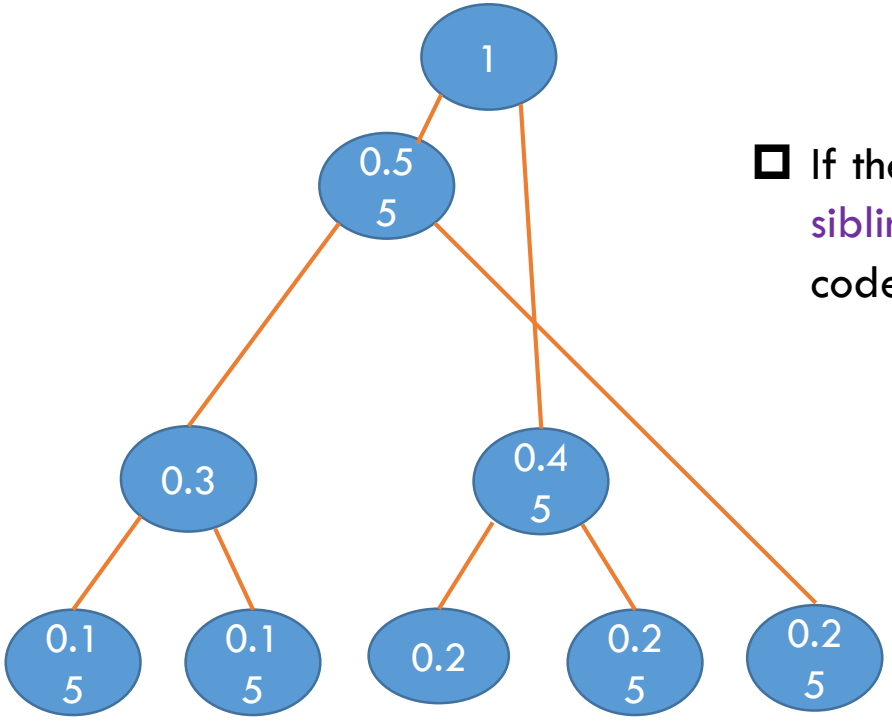


- If the two longest codewords are not of the same length, one can delete the last bit of the longer one, preserving the prefix property and achieving lower expected codeword length.

1. The lengths are ordered inversely with the probabilities (i.e., if $p_j > p_k$, then $l_j \leq l_k$).
2. The two longest codewords have the same length.

Canonical Codes: III

Without loss of generality, we will assume that the probability masses are ordered, so that $p_1 \geq p_2 \geq \cdots \geq p_m$. Recall that **a code is optimal if $\sum p_i l_i$ is minimal**. For any optimal coding scheme



- If there is a maximal-length codeword without a sibling(兄弟姐妹), we can delete the last bit of the codeword and still satisfy the prefix property

1. The lengths are ordered inversely with the probabilities (i.e., if $p_j > p_k$, then $l_j \leq l_k$).
2. The two longest codewords have the same length.
3. Two of the longest codewords differ only in the last bit and correspond to the two least likely symbols.

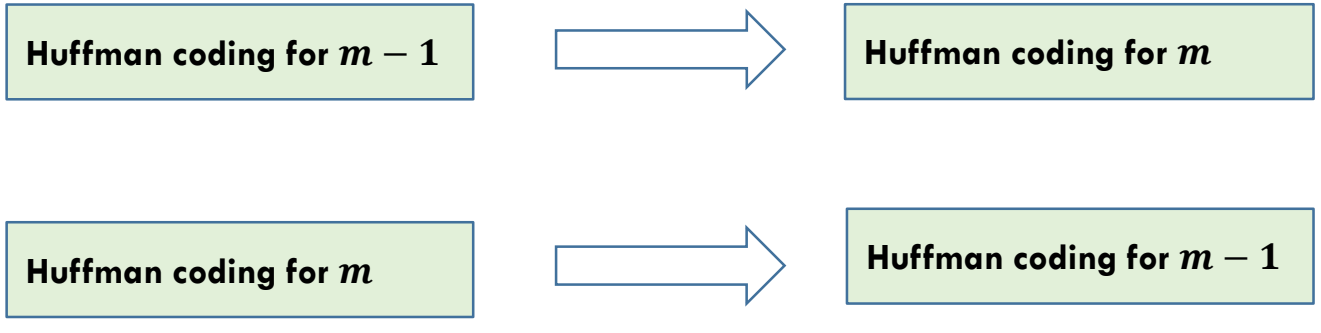
Optimality: Strategy

We prove the optimality of Huffman coding for a binary alphabet

- When $m = 2$, it is trivial
- For any probability mass function for an alphabet of size m, $\mathrm{p} = (p_1, p_2, \ldots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define $\mathrm{p}' = (p_1, p_2, \ldots, p_{m-2}, p_{m-1} + p_m)$ over an alphabet of size $m - 1$.

  Now we need to prove the optimality Huffman coding on p by the Huffman code on p'

  Challenge: Not so obvious



Optimality: $m-1 \rightarrow m$

- For any probability mass function for an alphabet of size m, $\mathrm{p} = (p_1, p_2, \ldots, p_m)$ with $p_1 \geq p_2 \geq \cdots \geq p_m$, we define $\mathrm{p}' = (p_1, p_2, \ldots, p_{m-2}, p_{m-1} + p_m)$ over an alphabet of size $m-1$.
- Let $C^*_{m-1}(\mathrm{p}')$ be an optimal code for p'. Let $C_m(\mathrm{p})$ be a code for p

$$C^*_{m-1}(\mathrm{p}') \Rightarrow C_m(p)$$

	$C^*_{m-1}(\mathbf{p}')$		$C_m(\mathbf{p})$	
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}	$w_{m-1} = w'_{m-1}0$	$l_{m-1} = l'_{m-1} + 1$
			$w_m = w'_{m-1}1$	$l_m = l'_{m-1} + 1$

- Expand an optimal code for p' to construct a code for p

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

(L and L^*)

$C_m(p)$ is a Huffman code. Maybe not optimal

Optimality: $m \rightarrow m-1$

From the canonical code for $\mathbf{p}$, we construct a code for $\mathbf{p}'$ by merging the codewords for the two lowest-probability symbols $m-1$ and m with probabilities p_{m-1} and p_m, which are siblings by the properties of the canonical code. The new code for $\mathbf{p}'$ has average length:

$$\begin{aligned} L(\mathbf{p}') &= \sum_{i=1}^{m-2} p_i l_i + p_{m-1}(l_{m-1}-1) + p_m(l_m-1) \\ &= \sum_{i=1}^{m} p_i l_i - p_{m-1} - p_m \\ &= L^*(\mathbf{p}) - p_{m-1} - p_m. \end{aligned}$$

- Expand an optimal code for p' to construct a code for p

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

- Condense an optimal canonical code for p to construct a code for the reduction p'

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

- Together,

$$L(\mathrm{p}) + L(\mathrm{p}') = L^*(\mathrm{p}) + L^*(\mathrm{p}')$$

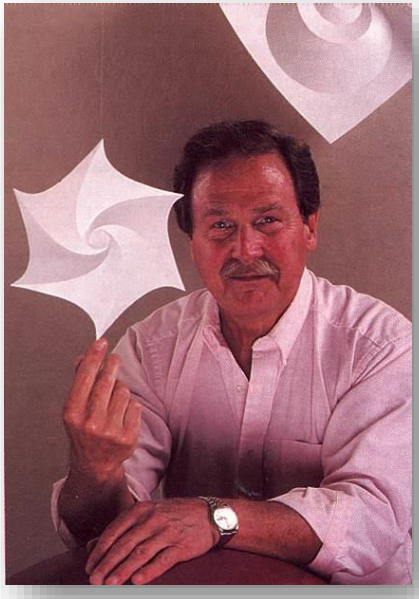
Since $L(\mathrm{p}) \geq L^*(\mathrm{p})$, $L(\mathrm{p}') \geq L^*(\mathrm{p}')$

$$\boldsymbol{L(\mathrm{p}) = L^*(\mathrm{p})} \textbf{ and } \boldsymbol{L(\mathrm{p}') = L^*(\mathrm{p}')}$$

- **Let the optimal code on p' be a Huffman code, then the expanded code on p is also a Huffman code and it is optimal for p.**

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

David Huffman



David Albert Huffman
(1925 – 1999)

He then served in the U.S. Navy as a radar maintenance officer on a destroyer that helped to **clear mines in Japanese and Chinese waters after World War II**.

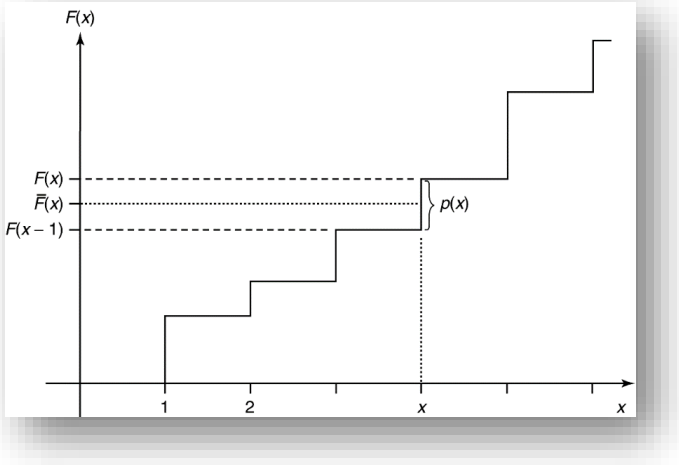
"Huffman code is one of the fundamental ideas that people in computer science and data communications are using all the time"
Donald E. Knuth, Stanford University

- Huffman worked on the problem for months, developing a number of approaches, but none that he could prove to be the most efficient. **Finally, he despaired of ever reaching a solution and decided to start studying for the final. Just as he was throwing his notes in the garbage, the solution came to him.** "It was the most singular moment of my life," Huffman says. "There was the absolute lightning of sudden realization."
- Huffman says he might never have tried his hand at the problem—much less solved it at the age of 25—if he had known that Fano, his professor, and Claude E. Shannon, the creator of information theory, had struggled with it. "It was my luck to be there at the right time and also **not have my professor discourage me by telling me that other good people had struggled with this problem,**" he says.
- https://www.huffmancoding.com/my-uncle/scientific-american

Shannon–Fano–Elias Coding

- Motivation: the codeword lengths $l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil \Rightarrow$ Kraft's inequality
- Without loss of generality, we can take $\mathcal{X} = \{1, 2, \ldots, m\}$. Assume that $p(x) > 0$ for all x. The **cumulative(累积) distribution function $F(x)$** is defined as $F(x) = \sum_{a \le x} p(a)$.
- Consider the modified cumulative distribution function

$$\bar{F}(x) = \sum_{a<x} p(a) + \frac{1}{2}p(x) = F(x) - \frac{1}{2}p(x)$$



- The step size is $p(x)$. $\bar{F}(x)$ is the midpoint
- $\bar{F}(x)$ can determine x. Thus is a code for x

- $\bar{F}(x)$ is a real number. Truncate $\bar{F}(x)$ to $l(x)$ bits and use the first $l(x)$ bit of $\bar{F}(x)$ as a code for x. Denote by $\lfloor \bar{F}(x) \rfloor_{l(x)}$.
- We have: $\bar{F}(x) - \lfloor \bar{F}(x) \rfloor_{l(x)} \le \frac{1}{2^{l(x)}}$
- If $l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1$,

$$\frac{1}{2^{l(x)}} \le \frac{p(x)}{2} = \bar{F}(x) - \bar{F}(x-1)$$

- $\lfloor \bar{F}(x) \rfloor_{l(x)}$ lies within the step corresponding to x. Thus, $l(x)$ bits suffice to describe x. (Prefix-free code)

$$L = \sum p(x)l(x) < H(X) + 2$$

$$p(x) \Rightarrow F(x) = \sum_{a \le x} p(a) \Rightarrow \bar{F}(x) = F(x) - \frac{1}{2}p(x) \Rightarrow l(x) + 1 \text{ bits}$$

Shannon–Fano–Elias Coding

x	$p(x)$	$F(x)$	$\overline{F}(x)$	$\overline{F}(x)$ in Binary	$l(x) = \left\lceil \log \frac{1}{p(x)} \right\rceil + 1$	Codeword
1	0.25	0.25	0.125	0.001	3	001
2	0.5	0.75	0.5	0.10	2	10
3	0.125	0.875	0.8125	0.1101	4	1101
4	0.125	1.0	0.9375	0.1111	4	1111

$$p(x) \Rightarrow F(x) = \sum_{a \le x} p(a) \Rightarrow \overline{F}(x) = F(x) - \frac{1}{2}p(x) \Rightarrow l(x) + 1 \text{ bits}$$

The average codeword length is 2.75 bits and the entropy is 1.75 bits. The Huffman code for this case achieves the entropy bound.

- Direct application of Shannon–Fano–Elias coding would also need arithmetic whose precision grows with the block size, which is not practical when we deal with long blocks.
- Shannon–Fano–Elias ⇒ Arithmetic coding

(Optimality) Let $l(x)$ be the codeword lengths associated with the Shannon code, and let $l'(x)$ be the codeword lengths associated with any other uniquely decodable code. Then

$$\Pr(l(X) \ge l'(X) + c) \le \frac{1}{2^{c-1}}$$

Hence, no other code can do much better than the Shannon code most of the time.

Summary

Cover: 5.6, 5.7, 5.8, 5.9, 5.10