Information and Communication Theory

Lecture 3

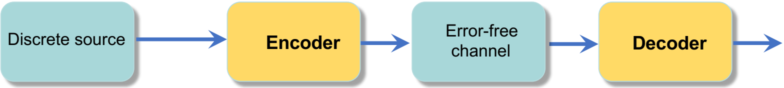
Optimal Coding

Mário A. T. Figueiredo

DEEC, Instituto Superior Técnico, University of Lisbon, **Portugal**

2023

Source Coding

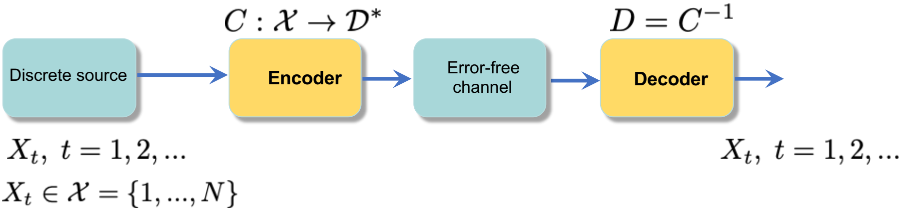


$X_t,\ t = 1, 2, \ldots$ $\qquad$ $X_t,\ t = 1, 2, \ldots$

$X_t \in \mathcal{X} = \{1, \ldots, N\}$

- Lossless encoding: output of the decoder equal to that of the source.
- Assumption: when encoding X_t, its distribution is known:
  - ✓ For memoryless sources, this is just f_X;
  - ✓ For Markov sources, this is $f_{X_t|X_{t-1},\ldots,}$
- Without loss of generality, we simply write f_X.
- Goal: economy, that is, use the channel as little as possible.

Variable-Length Coding



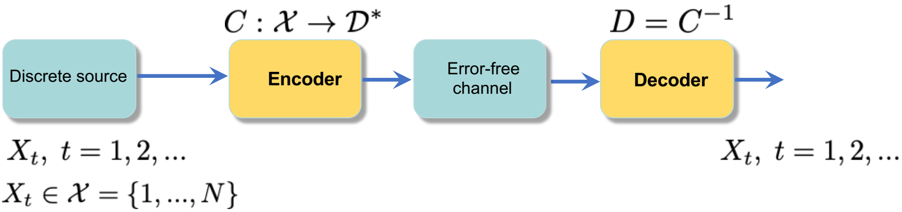
- Code uses D-ary alphabet $\mathcal{D} = \{0, 1, \ldots, D - 1\}$.
- Typically, binary coding, $D = 2$, $\mathcal{D} = \{0, 1\}$.
- Variable-length encoding: $\mathcal{D}^*$ is the Kleene closure of $\mathcal{D}$:

$$\mathcal{D}^* = \{\text{all finite strings of symbols of } \mathcal{D}\} = \bigcup_{n=0}^{\infty} \mathcal{D}^n$$

- Example: for $\mathcal{D} = \{0, 1\}$,

$$\mathcal{D}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \ldots\}$$

Non-Singular and Uniquely Decodable Codes



- For C^{-1} to exist: non-singular code (C injective). For any $x, y \in \mathcal{X}$,

$$x \neq y \ \Rightarrow C(x) \neq C(y)$$

- To be useful for a sequence of symbols, this is not good enough.

  Example: $\{C(1) = 0,\ C(2) = 10,\ C(3) = 01\}$ is non-singular

  Received sequence: 010; is it $C(1)C(2)$ or $C(3)C(1)$?

  Impossible to know!

- Codes that do not have this problem are called uniquely decodable.

Instantaneous Codes

- Consider a uniquely decodable code:
  $\{C(1) = 01, C(2) = 11, C(3) = 00, C(4) = 110\}$
- How to decode the sequence $11\underbrace{00....00}_{n \text{ zeros}}11$?

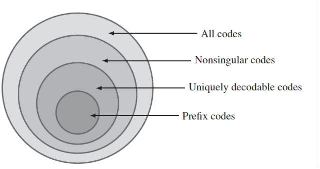
  ✓ If n is even: $C^{-1}(11\underbrace{00....00}_{n \text{ zeros}}11) = 2\,\underbrace{3....3}_{n/2}\,2$

  ✓ If n is odd: $C^{-1}(11\underbrace{00....00}_{n \text{ zeros}}11) = 4\,\underbrace{3....3}_{\frac{n-1}{2}}\,2$

- To decode the first symbol, we many need to wait for many others.
- A code that does not have this problem is called instantaneous.

Instantaneous Codes

- If no codeword is prefix of another, decoding is instantaneous. Other names: prefix codes, prefix-free codes.



- Length function:

$$l_C(x) = \text{length}(C(x))$$

- Expected length

$$L(C) = \mathbb{E}[l_C(X)] = \sum_{x \in \mathcal{X}} f_X(x)\, l_C(x)$$

- Example:

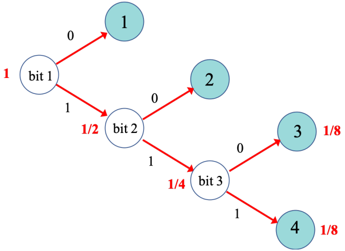
x	$f_X(x)$	C(x)	$l_C(x)$
1	1/2	0	1
2	1/4	10	2
3	1/8	110	3
4	1/8	111	3

$$L(C) = 7/4 \text{ bits/symbol}$$

Instantaneous Codes: Tree Representation

- Instantaneous code: no codeword is prefix of another.
- Decoding instantaneous code: path from root to leaf of a tree:

x	$f_X(x)$	C(x)	$l_C(x)$
1	1/2	0	1
2	1/4	10	2
3	1/8	110	3
4	1/8	111	3



- For D-ary codes: $D-$ary trees.
- $L(C)$ is the sum of the probabilities of the inner nodes. (show why)

Instantaneous Codes: Kraft-McMillan Inequality

- If C is a D-ary instantaneous code, it necessarily satisfies

$$\sum_{x\in\mathcal{X}} D^{-l_C(x)} \leq 1 \qquad \text{(KMI)}$$

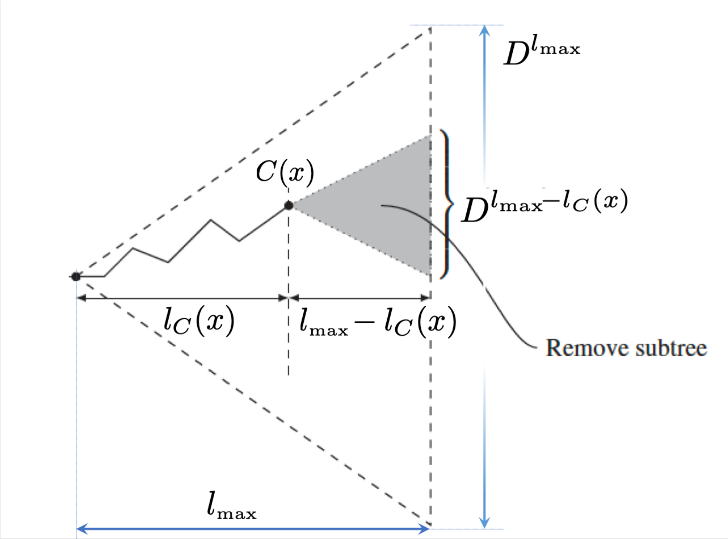
- ...i.e., if some words are short others have to be long!

- Proof:
  - ✓ let $l_{\max} = \max\{l_C(1), ..., l_C(N)\}$ (length of the longest word).
  - ✓ there are $D^{l_{\max}}$ words of length $l_{\max}$.
  - ✓ for each word $C(x)$, there are $D^{l_{\max}-l_C(x)}$ words of length $l_{\max}$ that have $C(x)$ as prefix;
  - ✓ the sets of length-$l_{\max}$ words that have each word as prefix are disjoint.

$$\sum_{x\in\mathcal{X}} D^{l_{\max}-l_C(x)} \leq D^{l_{\max}} \xrightarrow{\text{divide by } D^{l_{\max}}} \sum_{x\in\mathcal{X}} D^{-l_C(x)} \leq 1$$

- Important: the KMI is a necessary, not sufficient, condition. (why?)

Kraft-McMillan Inequality: Graphical Proof



Source Coding Theorem

- Source $X \in \mathcal{X} = \{1, ..., N\}$ with probability mass function f_X.
- For any collection of N positive integers, $l_1, ..., l_N$,

$$\text{(KMI)} \quad \sum_{x \in \mathcal{X}}^{N} D^{-l_x} \leq 1 \;\; \Rightarrow \;\; \sum_{x \in \mathcal{X}} f_X(x)\, l_x \geq H(X).$$

- Proof: let $q(x) = \dfrac{D^{-l_x}}{A} > 0$, where $A = \sum_{x \in \mathcal{X}} D^{-l_x} \leq 1$; $\sum_{x \in \mathcal{X}} q(x) = 1$

$$\begin{aligned} 0 \leq D_{\mathsf{KL}}(f_X \parallel q) &= \sum_{x \in \mathcal{X}} f_X(x) \log_D \frac{f_X(x)}{q(x)} \\ &= \underbrace{\sum_{x \in \mathcal{X}} f_X(x) \log_D f_X(x)}_{-H_D(X)} + \underbrace{\log_D A}_{\leq 0} \sum_{x \in \mathcal{X}} f_X(x) + \sum_{x \in \mathcal{X}} f_X(x)\, l_x \end{aligned}$$

...equality iff $A = 1$ and $q(x) = f_X(x) \Leftrightarrow l_x = -\log_D f_X(x)$ (only possible if integers).

Source Coding Theorem

- Source $X \in \mathcal{X} = \{1, ..., N\}$ with probability mass function f_X.
- Corollary of the result in previous slide:

$$C \text{ is instantaneous } \Rightarrow \text{ KMI } \Rightarrow \underbrace{\sum_{x \in \mathcal{X}} f_X(x)\, l_C(x)}_{\substack{\text{expected} \\ \text{code-length } L(C)}} \geq H_D(X)$$

- ...with equality if and only if $l_C(x) = -\log f_X(x)$.
- Equality is only possible if $-\log_D f_X(x)$ are integers.
- Shannon-Fano code (SFC): just take $l_C^{\mathsf{SF}}(x) = \lceil -\log_D f_X(x) \rceil$
- Clearly, the SFC satisfies the KMI ($\lceil u \rceil \geq u$, for any $u \in \mathbb{R}$)

$$\sum_{x \in \mathcal{X}} D^{-l_C^{\mathsf{SF}}(x)} \leq \sum_{x \in \mathcal{X}} D^{\log_D f_X(x)} = \sum_{x \in \mathcal{X}} f_X(x) = 1$$

Optimal Code

- Source $X \in \mathcal{X} = \{1, ..., N\}$ with probability mass function f_X.
- Optimal code lengths:

$$\left(l_1^{\text{optimal}}, ..., l_N^{\text{optimal}}\right) = \arg\min_{(l_1,...,l_N)} \sum_{x\in\mathcal{X}} f_X(x)\, l_x$$

$$\text{subject to } \; l_1, ..., l_N \in \mathbb{N}$$

$$\sum_{x\in\mathcal{X}} D^{-l_x} \leq 1$$

- Optimal code: C^{optimal} is any instantaneous code with

$$l_{C^{\text{optimal}}}(x) = l_x^{\text{optimal}}, \;\; \text{for } x \in \mathcal{X}$$

- Because it satisfies the KMI, $L(C^{\text{optimal}}) \geq H(X)$.
- Because it is optimal, $L(C^{\text{optimal}}) \leq L(C^{\text{SF}})$

Bounds on Optimal Code-length

- Because it is optimal, $L(C^{\text{optimal}}) \leq L(C^{\text{SF}})$
- Because $\lceil u \rceil < u + 1$, for any $u \in \mathbb{R}$,

$$L(C^{\text{optimal}}) \leq L(C^{\text{SF}}) = \sum_{x \in \mathcal{X}} f_X(x) \lceil -\log_D f_X(x) \rceil$$
$$< \sum_{x \in \mathcal{X}} f_X(x) (-\log_D f_X(x) + 1) = H(X) + 1$$

- In summary: $H(X) \leq L(C^{\text{optimal}}) < H(X) + 1$
- Code efficiency: $\rho_C = \dfrac{H(X)}{L(C)}$.
- Ideal code: $\rho_C = 1$. Important: ideal $\underset{\not\Leftarrow}{\Rightarrow}$ optimal

Coding With a Wrong Distribution

- Source $X \in \mathcal{X} = \{1, ..., N\}$ with probability mass function f_X.
- Build Shannon-Fano code assuming g_X: $l_C(x) = \lceil -\log g_X(x) \rceil$
- Lower bound:

$$
\begin{aligned}
L(C) &= \sum_{x \in \mathcal{X}} \lceil -\log g_X(x) \rceil f_X(x) \\
&\geq -\sum_{x \in \mathcal{X}} f_X(x) \log g_X(x) \\
&= \sum_{x \in \mathcal{X}} f_X(x) \log \frac{f_X(x)}{g_X(x)\, f_X(x)} \\
&= \underbrace{-\sum_{x \in \mathcal{X}} f_X(x) \log f_X(x)}_{H(X)} + \underbrace{\sum_{x \in \mathcal{X}} f_X(x) \log \frac{f_X(x)}{g_X(x)}}_{D_{\mathsf{KL}}(f_X \| g_X)}
\end{aligned}
$$

Coding With a Wrong Distribution

- Source $X \in \mathcal{X} = \{1, ..., N\}$ with probability mass function f_X.
- Build Shannon-Fano code assuming g_X: $l_C(x) = \lceil -\log g_X(x) \rceil$
- Upper bound:

$$
\begin{aligned}
L(C) &= \sum_{x \in \mathcal{X}} \lceil -\log g_X(x) \rceil f_X(x) \\
&< \sum_{x \in \mathcal{X}} f_X(x)(-\log g_X(x) + 1) \\
&= H(X) + D_{\mathsf{KL}}(f_X \parallel g_X) + 1
\end{aligned}
$$

- Summarizing: if C is built from g_X and the true distribution is f_X

$$
H(X) + D_{\mathsf{KL}}(f_X \parallel g_X) \leq L(C) < H(X) + D_{\mathsf{KL}}(f_X \parallel g_X) + 1
$$

Approaching the Bound: Source Extension

- Discrete stationary source $X_t \in \mathcal{X} = \{1, ..., N\}$
- Extension: group n consecutive symbols: $(X_1, ..., X_n) \in \{1, ..., N\}^n$.
- The optimal code for the extended symbols $(X_1, ..., X_n)$ satisfies

$$H(X_1, ..., X_n) \le \underbrace{L\left(C_n^{\text{optimal}}\right)}_{\text{bits/}(n\text{ symbols})} < H(X_1, ..., X_n) + 1$$

- Memoryless source: $H(X_1, ..., X_n) = n\,H(X_1)$, thus

$$L\left(C_n^{\text{optimal}}\right) \le n\,H(X_1) + 1 \quad \Rightarrow \quad \underbrace{\frac{L\left(C_n^{\text{optimal}}\right)}{n}}_{\text{bits/symbol}} < H(X_1) + \frac{1}{n}$$

  ...via extension, expected code-length can arbitrarily approach the entropy.

- Non-memoryless source: $H(X_1, ..., X_n) < nH(X_1)$ and the result is even stronger.

Huffman Codes

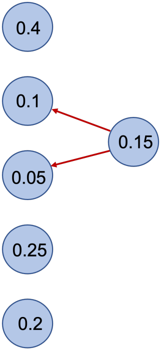
- Huffman (1952) algorithm to obtain optimal codes.
- Builds a $D-$ary tree, starting from the leaves, which are the symbols.
- Algorithm (for $D = 2$; the generalizing to $D > 2$ requires some care).
  1. **Input**: a list of symbol probabilities $(p_1, ..., p_N)$.
  2. **Output**: a binary tree with each symbol as a leaf.
  3. Assign each symbol to a leaf of the tree.
  4. Find the 2 smallest probabilities: p_i and p_j.
  5. Create the parent node for nodes i and j with probability $p_i + p_j$.
  6. Remove p_i and p_j from the list and insert $p_i + p_j$.
  7. If the list of symbols has more than 2 probabilities, go back to step 4.
- As seen before, a binary tree corresponds to an instantaneous code.

Huffman Codes

- Illustration: probabilities $(0.4, 0.1, 0.05, 0.25, 0.2)$

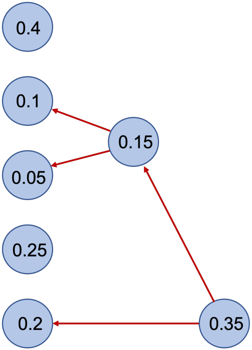
Huffman Codes

- Illustration: probabilities $(0.4, 0.1, 0.05, 0.25, 0.2)$



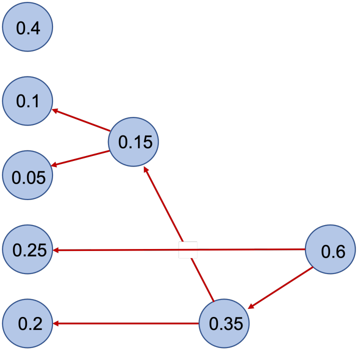
Huffman Codes

- Illustration: probabilities $(0.4, 0.1, 0.05, 0.25, 0.2)$



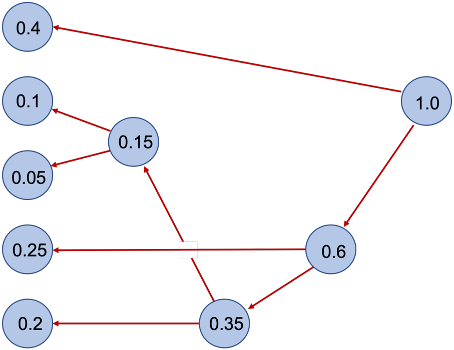
Huffman Codes

- Illustration: probabilities $(0.4, 0.1, 0.05, 0.25, 0.2)$



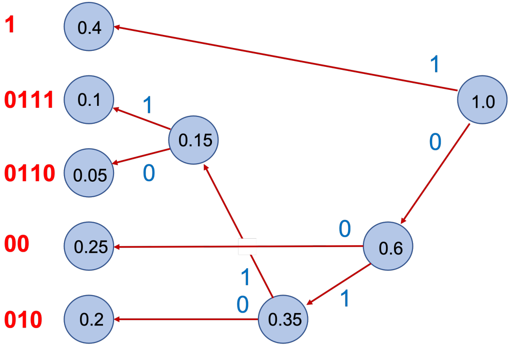
Huffman Codes

- Illustration: probabilities $(0.4, 0.1, 0.05, 0.25, 0.2)$



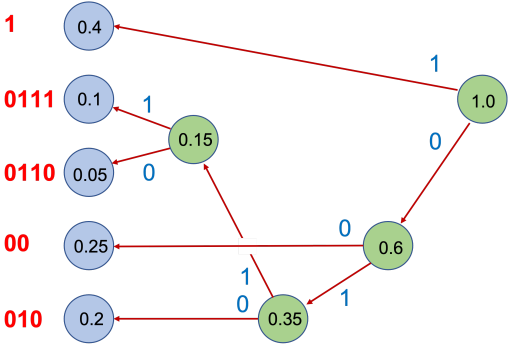
Huffman Codes

- Label the edges (arbitrarily) to obtain the code words



Huffman Codes

- Expected code-length: sum of the inner node probabilities:
  $L(C) = 1 + 0.6 + 0.35 + 0.15 = 2.1$ bits/symbol



Huffman Codes

- Huffman codes are optimal; see proof in recommended reading.

- Converse is not true

$$\text{Huffman code} \;\; \underset{\not\Leftarrow}{\Rightarrow} \;\; \text{optimal code}$$

- In the case of ties, break them arbitrarily.

- For D-ary codes, merge D symbols to build a D-ary tree.

- For D-ary codes, optimality requires $N = k(D-1)+1$, where $k \in \mathbb{N}$.
  ...if not satisfied, just append zero-probability symbols.

Elias Codes for Natural Numbers

- Standard number representation is not uniquely decodable.
- Binary representation of natural numbers is not uniquely decodable. Example: $C(3) = 11$, $C(21) = 10101$, but decoding 1110101 is impossible; it could be $C(14)C(5)$ or $C(58)C(1)$.
- Length of binary representation for $x \in \mathbb{N}$ is $\lfloor \log_2 x \rfloor + 1$. Example: $C(13) = 1101$ has length 4; $\lfloor \log_2 13 \rfloor + 1 = \lfloor 3.70 \rfloor + 1 = 4$.
- Elias coding:
  - ✓ instantaneous code for arbitrary natural numbers;
  - ✓ length not much worse than $\lfloor \log_2 x \rfloor + 1$.
- Useful not only for $\mathbb{N}$, but also for large alphabets $\mathcal{X} = \{1, ..., N\}$ with large and unknown N.

Elias Gamma Code

- Length of binary representation for $x \in \mathbb{N}$ is $\lfloor \log_2 x \rfloor + 1$.
- Let C_2 denote the standard binary representation.
- Elias gamma code:

$$C_\gamma(x) = \underbrace{0\ldots0}_{\lfloor \log_2 x \rfloor \text{zeros}} C_2(x)$$

x	$C_\gamma(x)$
1	1
2	010
4	00100
5	00101
7	00111
9	0001001
10	0001010
⋮	⋮
19	000010011
⋮	⋮
147	000000010010011

- Obviously instantaneous.
- Length:
  $l_{C_\gamma}(x) = 2\lfloor \log_2 x \rfloor + 1$.
- Twice as long as C_2.

Elias Delta Code

- Elias delta code: $C_\delta(x) = C_\gamma(\lfloor \log_2 x \rfloor + 1)\ \tilde{C}_2(x)$
- $\tilde{C}_2$ is C_2 without the leading 1 (e.g. $C_2(9) = 1001,\ \tilde{C}_2(10) = 001$)
- Length: $$l_{C_\delta}(x) = l_{C_\gamma}(\lfloor \log_2 x \rfloor + 1) + \lfloor \log_2 x \rfloor$$ $$= 2\lfloor \log_2(\lfloor \log_2 x \rfloor + 1) \rfloor + \lfloor \log_2 x \rfloor + 1$$

x	$C_\delta(x)$
1	1
2	0100
3	0101
4	01100
7	01111
8	00100000
10	00100010
⋮	⋮
19	001010011
⋮	⋮
147	0001000010011

- Obviously instantaneous.
- For $x > 32$, $l_{C_\delta}(x) < l_{C_\gamma}(x)$
- Approaches C_2 for large x:

$$\lim_{x\to\infty} \frac{l_{C_\delta}(x)}{C_2(x)} = 1$$

Recommended Reading

- T. Cover and J. Thomas, "Elements of Information Theory", John Wiley & Sons, 2006 (Chapter 5).
- M. Figueiredo, "Elias Coding for Arbitrary Natural Numbers", available at the course webpage in Fenix.