ECE253/CSE208 Introduction to Information Theory

Lecture 7: Data Compression: Prefix Code & Kraft-McMillan Inequality

Dr. Yu Zhang
ECE Department
University of California, Santa Cruz

- Chap 5 of *Elements of Information Theory (2nd Edition) by Thomas Cover & Joy Thomas*

Different classes of codes

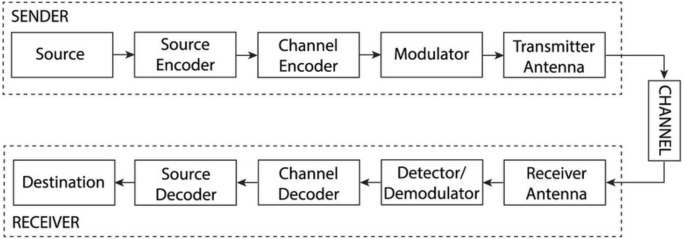

Figure: A Diagram of Communication Systems.

- Chapters 2–4 describe fundamental characteristics of information sources.
- Source coding compresses the source (redundancy reduction) as much as possible without losing information.
- Channel coding combats channel noise to control errors (by introducing redundancy).
- **Data compression**: Encoding the given source symbols into strings (codewords)
- Entropy of a random source is the fundamental limit of information compression.

Source Code

Definition

A source code $C(x) : \mathcal{X} \to D^*$, where $\mathcal{D}^*$ is the set of finite length strings of symbols from D-ary alphabet; e.g. $\mathcal{D} := \{0, 1, \cdots, D-1\}$.

Example

Consider $\mathcal{X} = \{1, 2, 3, 4\}$, one possible coding scheme:

$$C(1) = 0,\ C(2) = 10,\ C(3) = 110,\ C(4) = 111$$

Definition (Nonsingular code)

$x \neq x' \implies c(x) \neq c(x')$. We can add a special symbol (e.g., comma) between any two consecutive codewords for decoding, but this is inefficient.

Definition

Extension C^* of a code C: Concatenation of the corresponding codewords

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

Source Code (Cont'd)

Definition (Uniquely decodable)

If a code's extension is non-singular, then a code is uniquely decodable. In other words, a uniquely decodable code has only one source string producing it. However, we may need to look at the entire string to determine the source.

Sardinas-Patterson algorithm (1953'): A classical algorithm for determining in polynomial time whether a given variable-length code is uniquely decodable.

Definition (Prefix code)

A code is called a prefix (or instantaneous) code if no codeword is a prefix of any other codewords. A prefix code can be decoded without reference to future codewords since the end of a codeword is immediately recognizable (self-punctuating).

Source Code (Cont'd)

X	Singular	Nonsingular, but not Uniquely Decodable	Uniquely Decodable, but not Instantaneous	Instantaneous
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

Figure: Classes of codes: The 2nd class is not uniquely decodable because the codeword 010 can be decoded as 2, 31, or 14. The 3rd class is not a prefix code because 11 is a prefix of 110.

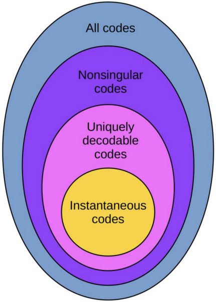

Figure: A diagram showing relations of different classes of codes.

Kraft-McMillan Inequality

Theorem (Kraft inequality)

For any prefix code over an alphabet of size D, the codeword length $l_1, l_2, \ldots, l_m$ must satisfy the following inequality:

$$\boxed{\sum_{i=1}^{m} D^{-l_i} \leq 1.}$$

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

Extended Kraft Ineq: the inequality holds for an infinite set of prefix code $(m \to \infty)$.

Exponentiated codeword length assignments must look like a PMF.

Insight. Kraft inequality shows a budget constraint of codeword lengths for any prefix codes. To minimize the expected code length, we would like to assign shorter codewords to more frequent symbols. However, *shorter codewords are more expensive.*

Kraft-McMillan Inequality (Cont'd)

Implications.

- If Kraft's inequality does not hold, the code is not uniquely decodable.
- If Kraft's inequality holds with strict inequality, the code is called *redundant*.
- If Kraft's inequality holds with equality, the code is called *complete*.
- For any redundant prefix code with codeword lengths $\{l_i\}_{i=1}^m$ there exists a complete prefix code with codeword lengths $\{l'_i\}_{i=1}^m$ such that $l'_i \leq l_i$ for all $i = 1, 2, \ldots, m$.

Proof of Kraft Inequality

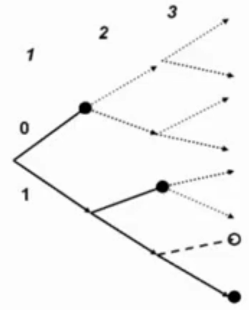

Figure: Constructing a binary tree. Three black nodes denote different codewords. The branches represent the symbols of the codeword. The path from the root traces out the symbols of the codeword.

Proof.

- Prefix condition: no codeword is an ancestor of any other codewords (each codeword eliminates its descendants as possible codewords).
- Let $l_{\max}$ be the length to the longest codeword.
- The leaf nodes at level $l_{\max}$ can be codewords, descendants, or unused.
- A codeword at level l_i has $2^{(l_{\max}-l_i)}$ descendants at level $l_{\max}$.
- Each of these descendant sets must be disjoint.
- Summing over all codewords, we have $\sum_i 2^{(l_{\max}-l_i)} \leq 2^{l_{\max}} \implies \sum_i 2^{-l_i} \leq 1$.
- Clearly, the argument holds for an arbitrary D-ary tree.

$\square$

McMillan Inequality

The McMillan inequality generalizes the Kraft inequality from *prefix code* to *uniquely decodable code*:

Theorem (McMillan inequality)

For any uniquely decodable code over an alphabet of size D, the codeword length $l_1, l_2, \ldots, l_m$ must satisfy the following inequality: $\sum_i D^{-l_i} \leq 1$.
Conversely, give a set of codeword lengths that satisfy the inequality, it is possible to construct a uniquely decodable code with those codeword lengths.

Kraft-McMillan Inequality (Cont'd)

X	Singular	Nonsingular, but not Uniquely Decodable	Uniquely Decodable, but not Instantaneous	Instantaneous
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

Example (Sanity check)

- The prefix code in the table satisfies the Kraft inequality:

$$\sum_i D^{-l_i} = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} < 1.$$

- For the not uniquely decodable code, its lengths violate the Kraft-McMillan inequality:

$$\sum_i D^{-l_i} = 2^{-1} + 2^{-3} + 2^{-2} + 2^{-2} = \frac{9}{8} > 1.$$

Lemma

For every uniquely decodable code, there exists a prefix code with the same length distribution.

Optimal Codes (Shortest Expected Length)

Finding the code lengths of optimal codes (prefix/uniquely decodable codes) can be formulated as the following constrained optimization problem:

$$\underset{\{l_i\}}{\text{minimize}} \quad \sum_i p_i l_i \tag{1}$$

$$\text{subject to} \quad \sum_i D^{-l_i} \le 1 \tag{2}$$

Consider the Lagrangian relaxation by introducing the multiplier λ:

$$L(\{l_i\}, \lambda) = \sum_i p_i l_i + \lambda \left(\sum_i D^{-l_i} - 1 \right).$$

Set the gradient to be zero, we get:

$$\frac{\partial L(\{l_i\}, \lambda)}{\partial l_i} = p_i - \lambda D^{-l_i} \ln D = 0 \quad \Longrightarrow \quad D^{-l_i} = \frac{p_i}{\lambda \ln D}.$$

We should have $\sum_i D^{-l_i^*} = 1$ (i.e., the ineq constraint must be binding at the optimum).

Hence, $\lambda = \frac{1}{\ln D} \implies p_i = D^{-l_i^*} \implies \boxed{l_i^* = -\log_D p_i}$ (if $-\log_D p_i$ is an integer).

Optimal Codes (Cont'd)

The optimal expected codeword length (i.e., the optimal value of the objective function):

$$L_i^* = \sum p_i l_i^* = -\sum p_i \log_D p_i = H_D(X).$$

Again, we see that the entropy is a natural measure of efficient description length.

Theorem

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

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

with equality if and only if $D^{-l_i} = p_i$.

Optimal codes (Cont'd)

Proof: Let $r_i := \frac{D^{-l_i}}{\sum_i D^{-l_i}}, c := \sum_i D^{-l_i}$, we have

$$
\begin{aligned}
L - H_D(X) &= \sum_i p_i l_i - \sum_i p_i \log_D \frac{1}{p_i} && (3)\\
&= \sum_i -p_i \log_D D^{-l_i} + \sum_i p_i \log_D p_i && (4)\\
&= \sum_i p_i \log_D \frac{p_i}{r_i} - \log_D c && (5)\\
&= D(\mathbf{p}||\mathbf{r}) + \log_D 1/c && (6)\\
&\geq 0 && (7)
\end{aligned}
$$

Optimal Codes (Cont'd)

Definition (D-adic distribution)

A distribution is called D-adic if each of the probabilities is equal to D^{-n} for some $n \in \mathbb{Z}_+$.

One way to find the optimal code: Find the D-adic distribution that is closest (in the sense of KL divergence) to the distribution of X. But, this is a hard problem.

Theorem (Bounds on the optimal code length)

The minimum expected codeword length per symbol satisfies

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

- If $\{X_i\}$ are i.i.d. $\Rightarrow H(X_1) \leq L_n < H(X_1) + \frac{1}{n} \Rightarrow L_n \to H(X_1)$ as $n \to \infty$
- If $\{X_i\}$ are non-i.i.d. $\Rightarrow L_n \to H(\mathcal{X})$ as $n \to \infty$

Wrong Code

Q: If the code is designed based on a wrong distribution (e.g., we have a wrong estimate of p_i), how much penalty will we pay?

A: Suppose the true distribution is $\{p_i\}$, but we design the codes according to $\{q_i\}$.

Theorem (Wrong code)

Let $l(x) = \lceil \log_D \frac{1}{q(x)} \rceil$ *while the true distribution is* $p(x)$*. Then, we have*

$$H_D(X) + D(p||q) \leq E_p[l(X)] < H_D(X) + 1 + D(p||q)$$

Clearly, if $p = q$ (no mismatch), $H_D(X) \leq E_p[l(X)] < H_D(X) + 1$.

Proof:

$$\begin{aligned} E_p[l(X)] &= \sum_i p_i \left\lceil \log_D \frac{1}{q_i} \right\rceil < \sum_i p_i \left(1 + \log_D \frac{1}{q_i} \right) \\ &= 1 + \sum_i p_i \log_D \frac{p_i}{q_i} + \sum_i p_i \log_D \frac{1}{p_i} \\ &= 1 + D(p||q) + H_D(X). \end{aligned}$$

Thank You!

Email: <zhangy@ucsc.edu>

Homepage: https://people.ucsc.edu/~yzhan419/