

CS258: Information Theory

Fan Cheng

Shanghai Jiao Tong University

[http://www.cs.sjtu.edu.cn/~chengfan/
chengfan@sjtu.edu.cn](http://www.cs.sjtu.edu.cn/~chengfan/chengfan@sjtu.edu.cn)

Spring, 2020

Outline

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

Data Compression: Existing Systems



蓟镇总兵戚继光在《练兵纪实》中讲：“凡无空心台之处，即以原墩充之，有空心台所相近百步之内者，俱以空心台充墩。大约相去一、二里，梆鼓相闻为一墩。”戚继光还制定了传烽之法，编成通俗顺口的《传烽歌》让守台官兵背诵熟记。烽火台是白天点狼粪，晚上燃柴草，白天烧狼粪用烟比较明显，晚上烧柴草靠火光报警。



International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	• —	U	• • —
B	• • • —	V	• — — •
C	— • • —	W	— • — •
D	— • — •	X	— • • —
E	• — — —	Y	— — • —
F	• • — •	Z	— — — •
G	• — • —		
H	• • • •		
I	• • — —		
J	• — — —		
K	— • — •	1	— — — —
L	• — — —	2	• • — — —
M	— — — —	3	• • • — —
N	— • — —	4	• • • • —
O	— — — •	5	• • • • •
P	• — — —	6	— • — — —
Q	— • — •	7	— • — • —
R	• — — •	8	— • — • •
S	• • — —	9	— — • — —
T	— — — •	0	— — — — •

Morse code: 'E', 'I', '5', '4'

gzip

- “gzip” is a file format and a software application used for file compression and decompression. “g” is from “GNU”
- The gzip format is used in HTTP compression: Chrome, IE, Firefox
- Linux: gzip command
- Python: gzip module

```
import gzip
s_in = b"Lots of content here"
s_out = gzip.compress(s_in)
```

Abraham Lempel and
Jacob Ziv: LZ77, LZ78

Examples of Codes

Let X be a random variable with the following distribution and codeword assignment:

$\Pr(X = 1) = 1/2,$	codeword $C(1) = 0$
$\Pr(X = 2) = 1/4,$	codeword $C(2) = 10$
$\Pr(X = 3) = 1/8,$	codeword $C(3) = 110$
$\Pr(X = 4) = 1/8,$	codeword $C(4) = 111$

- Without loss of generality, we can assume that the D -ary alphabet is $\mathcal{D} = \{0, 1, \dots, D - 1\}$.
- A source code (信源编码) C for a random variable X is a mapping from \mathcal{X} , the range of X , to D^* , the set of finite-length strings of symbols from a **D-ary (D元组)** alphabet. Let **$C(x)$** denote the **codeword** corresponding to x and let **$l(x)$** denote the length of $C(x)$.
- The **expected length $L(C)$** of a source code $C(x)$ for a random variable X with probability mass function $p(x)$ is given by
$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x)$$
 - What is $\min L(C)$
 - How to construct such an optimal code

$$H(X) = \frac{7}{4} \quad \text{and} \quad L(C) = \frac{7}{4}$$

Nonsingular Code

- A code is said to be **nonsingular** if every element of the range of X maps into a different string in D^* ; that is,

$$\blacksquare x \neq x' \Rightarrow C(x) \neq C(x') \text{ (单射)}$$

- The **extension** C^* of a code C is the mapping from finite length strings of X to finite-length strings of D , defined by

$$C(x_1x_2 \dots x_n) = C(x_1)C(x_2) \dots C(x_n),$$

where $C(x_1)C(x_2) \dots C(x_n)$ indicates concatenation of the corresponding codewords.

- If $C(x_1) = 00$ and $C(x_2) = 11$, then $C(x_1x_2) = 0011$.

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

-- In other words, any encoded string in a uniquely decodable code has only one possible source string producing it.

- A code is called a **prefix code**(前缀码) or an **instantaneous code**(即时码) if no codeword is a prefix of any other codeword.

--An instantaneous code can be decoded without reference to future codewords since the end of a codeword is immediately recognizable

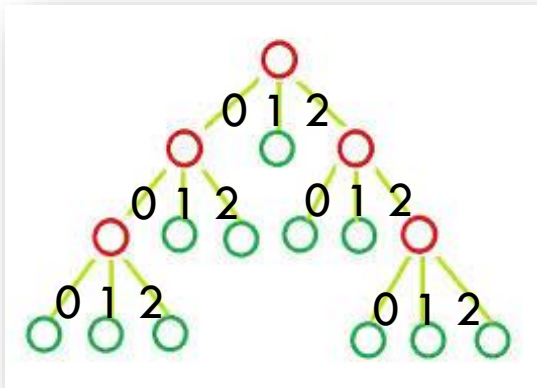
--Suffix code: no codeword is a suffix of any other codeword

How to construct a prefix code?
Enumerate?

Prefix Code: Representation

The space of D -ary prefix codes admits a very special structure. To represent them is of fundamental importance

- Tree representation
- Interval representation



Start from the root:

- Each node has D branches.
- Each edge corresponds one of $\{0, 1, \dots, D - 1\}$
- (Prefix-free) For two code words c_1, c_2 , the corresponding paths p_1, p_2 will not contain each other. (A subtree will be occupied)

$$\begin{array}{c} \xrightarrow{\quad [\quad] \quad} \\ [0.d_1d_2 \dots d_n, \quad 0.d_1d_2 \dots d_n + \frac{1}{D^n}) \end{array}$$

$$0.1211 \rightarrow 0.1211xxxx \rightarrow 0.1212$$

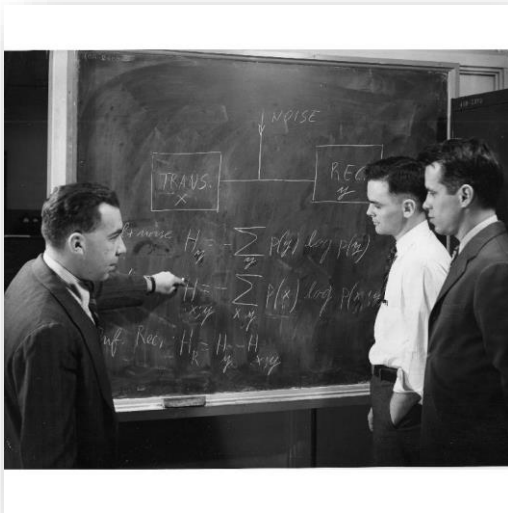
- Each codeword $d_1d_2 \dots d_n$ can be treated as a D -ary floating number $0.d_1d_2 \dots d_n$
- The left closed and right open interval $[0.d_1d_2 \dots d_n, 0.d_1d_2 \dots d_n + \frac{1}{D^n})$
- NO other codeword is allowed in it
- NO overlap between intervals

Kraft Inequality

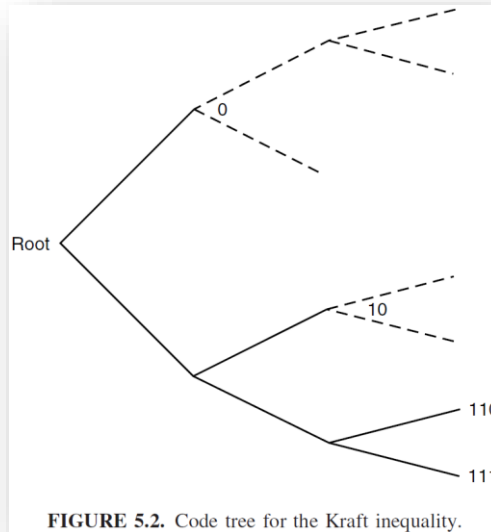
(Kraft Inequality 1949) For any instantaneous code (prefix code) over an alphabet of size D , the codeword lengths l_1, l_2, \dots, l_m must satisfy the inequality

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

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



Leon G. Kraft



D^{-l_i} can be treated as “percentage”

Assume $l_1 \leq l_2 \leq \dots \leq l_m$ (The maximum depth is l_m)

■ For l_i , it “occupied” a subtree in size $D^{l_m - l_i}$

■ The aggregate size of subtrees

$$\sum_{i=1}^m D^{l_m - l_i}$$

■

$$\sum_{i=1}^m D^{l_m - l_i} \leq D^{l_m} \Rightarrow \text{“only if”}$$

■ “if”: mathematical induction

Extended Kraft Inequality

(Extended Kraft Inequality) For **any countably infinite set of codewords** that form a prefix code, the codeword lengths satisfy the extended Kraft inequality,

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

Conversely, given any l_1, l_2, \dots satisfying the extended Kraft inequality, we can construct a prefix code with these codeword lengths.

- Let the D-ary alphabet be $\{0, 1, \dots, D - 1\}$. Consider the i th codeword $y_1 y_2 \dots y_{l_i}$. Let $0.y_1 y_2 \dots y_{l_i}$ be the real number given by the D-ary expansion

$$0.y_1 y_2 \dots y_{l_i} = \sum_{j=1}^{l_i} y_{l_j} D^{-j}$$

This codeword corresponds to the interval

$$\left[0.y_1 y_2 \dots y_{l_i}, 0.y_1 y_2 \dots y_{l_i} + \frac{1}{D^{l_i}} \right)$$

- “Floating number” is more general
- An equation to capture “prefix”

- ✓ This is a subinterval of the unit interval $[0, 1]$.
- ✓ By the prefix condition, these intervals are disjoint.
- Code construction: First, reorder the indexing so that $l_1 \leq l_2 \leq \dots$. Then simply assign the intervals in order **from the low end of the unit interval**.

Exercise

- (Slackness in the Kraft inequality.) An instantaneous code has word lengths l_1, l_2, \dots, l_m , which satisfy the strict inequality

$$\sum_{i=1}^m D^{-l_i} < 1$$

The code alphabet is $\mathcal{D} = \{0, 1, 2, \dots, D - 1\}$. Show that there exist arbitrarily long sequences of code symbols in D^* which cannot be decoded into sequences of codewords.

- (Fix-free codes) A code is a fix-free code if it is both a prefix code and a suffix code. Let l_1, l_2, \dots, l_m be m positive integers. Prove that if

$$\sum_{k=1}^m 2^{-l_k} \leq \frac{1}{2}$$

then there exists a binary fix-free code with codeword length l_1, l_2, \dots, l_m .

- ($\frac{3}{4}$ fix-free codes) Prove that when

$$\sum_{k=1}^m 2^{-l_k} \leq \frac{3}{4}$$

the conclusion above holds.

Optimal Codes: Problem Formulation

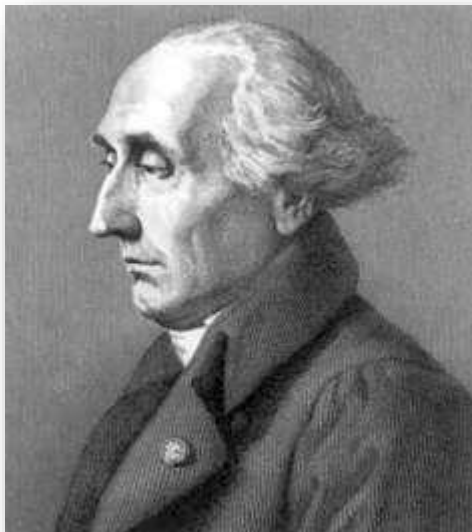
- **Kraft inequality** gives a mathematical expression on the existence of prefix code
- The problem of finding the prefix code with the minimum expected length could be formulated as a standard **optimization problem**

$$\min L = \sum p_i l_i$$

such that

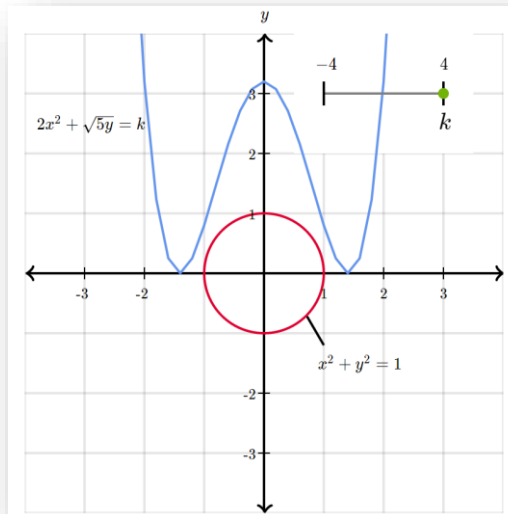
$$\sum D^{-l_i} \leq 1$$

- **How to solve it? Lagrange**



Joseph Louis Lagrange
1736--1813

$$\begin{array}{ll} \min f(X) \\ \text{s. t.} & g(X) \leq 0 \end{array}$$



- Their gradient vectors are parallel
 $\nabla f(X) = \lambda \nabla g$
- Lagrangian
 $\mathcal{L}(X, \lambda) = f(X) + \lambda g$
- Necessary condition
 $\nabla \mathcal{L} = 0$

Optimal Codes: Solution

- The Lagrange multipliers

$$J = \sum p_i l_i + \lambda (\sum D^{-l_i} - 1)$$

- Differentiating with respect to l_i , we obtain

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

- Setting the derivatives to 0, we obtain

$$D^{-l_i} = \frac{p_i}{\lambda \log_e D}$$

- Substituting this in the constraint to find λ , we find $\lambda = 1/\log_e D$, and hence

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

yielding optimal code lengths,

$$l_i^* = -\log_D p_i$$

- This noninteger choice of codeword lengths yields expected codeword length

$$L^* = \sum p_i l_i^* = \sum -p_i \log p_i = H_D(X)$$

- In general, $H_D(X)$ cannot be attained

$$L^* \geq H_D(X)$$

$$\begin{aligned} p_i &= D^{-l_i} \\ l_i &= -\log p_i \end{aligned}$$

Optimal Codes: Bounds

Let $l_1^*, l_2^*, \dots, l_m^*$ be optimal codeword lengths for a source distribution \mathbf{p} and D -ary alphabet, and let L^* be the associated expected length of an optimal code ($L^* = \sum p_i l_i^*$). Then

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

+1

- Recall that $p_i = D^{-l_i}$ and $l_i = -\log_D p_i$
- Since $\log_D \frac{1}{p_i}$ may not equal to an integer, we round it up to give integer word-length assignments,

$$l_i = \left\lceil \log_D \frac{1}{p_i} \right\rceil \Rightarrow \text{Shannon codes}$$

- Check l_i 's satisfying Kraft inequality.

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

- Take expectations

$$H_D(X) \leq L < H_D(X) + 1$$

Can we remove the “1” bit here?

Approach the Limit

- Encode n symbols X_1, X_2, \dots, X_n on \mathcal{X} together, where X_i 's are i.i.d $\sim p(x)$.
- Denote the alphabet by \mathcal{X}^n , the expected codeword length by L_n , the length of codeword associated with (x_1, x_2, \dots, x_n) by $l(x_1, x_2, \dots, x_n)$

$$L_n = \frac{1}{n} \sum p(x_1, x_2, \dots, x_n) l(x_1, x_2, \dots, x_n) = \frac{1}{n} El(X_1, X_2, \dots, X_n)$$

- **Treat X_1, X_2, \dots, X_n as a whole and apply the lower bound aforementioned**

$$H(X_1, X_2, \dots, X_n) \leq El(X_1, X_2, \dots, X_n) < H(X_1, X_2, \dots, X_n) + 1$$

- Since X_i 's are i.i.d, $H(X_1, X_2, \dots, X_n) = nH(X)$

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

(Theorem.) The minimum expected codeword length per symbol statisfies

$$\frac{H(X_1, X_2, \dots, X_n)}{n} \leq L^* < \frac{H(X_1, X_2, \dots, X_n)}{n} + \frac{1}{n}$$

Moreover, if X_1, X_2, \dots, X_n is a stationary stochastic process,

$$L^* \rightarrow H(\mathcal{X})$$

Entropy rate

Wrong code

What happens to the expected description length if the code is designed for the wrong distribution ($q(x)$). For example, the wrong distribution may be the best estimate that we can make of the unknown true distribution.

(Wrong code) The expected length under $p(x)$ of the code assignment $l(x) = \log \frac{1}{q(x)}$ satisfies

$$H(p) + D(p||q) \leq E_p l(x) < H(p) + D(p||q) + 1$$

The expected codelength is

$$\begin{aligned} El(x) &= \sum_x p(x) \left\lceil \log \frac{1}{q(x)} \right\rceil \\ &< \sum_x p(x) \left(\log \frac{1}{q(x)} + 1 \right) \\ &= \sum_x p(x) \log \frac{p(x)}{q(x)} \frac{1}{p(x)} + 1 \\ &= \sum_x p(x) \log \frac{p(x)}{q(x)} + \sum_x p(x) \log \frac{1}{p(x)} + 1 \\ &= D(p||q) + H(p) + 1 \end{aligned}$$

$$D(p||q)$$

Kraft Inequality For Uniquely Decodable Codes

(McMillan) The codeword lengths of **any uniquely decodable** D -ary code must satisfy the Kraft inequality

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

Conversely, given a set of codeword lengths that satisfy this inequality, it is possible to construct a uniquely decodable code with these codeword lengths.

- ✓ Consider C^k the k th extension of the code (i.e., the code formed by the concatenation of k repetitions of the given uniquely decodable code C).
- ✓ By the definition of unique decodability, the k th extension of the code is nonsingular.
- ✓ Since there are only D^n different D -ary strings of length n , unique decodability implies that the number of code sequences of length n in the k th extension of the code must be no greater than D^n .

The class of uniquely decodable codes **does not offer any** further possibilities for the set of codeword lengths than do instantaneous codes.

Kraft Inequality For Uniquely Decodable Codes (Cont'd)

- Let the codeword lengths of the symbols $x \in X$ be denoted by $l(x)$. For the extension code, the length of the code sequence is $l(x_1, x_2, \dots, x_k) = \sum_{i=1}^k l(x_i)$
- The inequality we wish to prove is $\sum_{x \in \mathcal{X}} D^{-l(x)} \leq 1$.
- Consider the k th power of this quantity

$$\left(\sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k = \sum_{x_1 \in \mathcal{X}} \sum_{x_2 \in \mathcal{X}} \dots \sum_{x_k \in \mathcal{X}} D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)}$$

$$= \sum_{x_1, x_2, \dots, x_k \in \mathcal{X}^k} D^{-l(x_1)} D^{-l(x_2)} \dots D^{-l(x_k)}$$

$$= \sum_{x^k \in \mathcal{X}^k} D^{-l(x^k)} = \sum_{m=1}^{kl_{\max}} a(m) D^{-m}$$

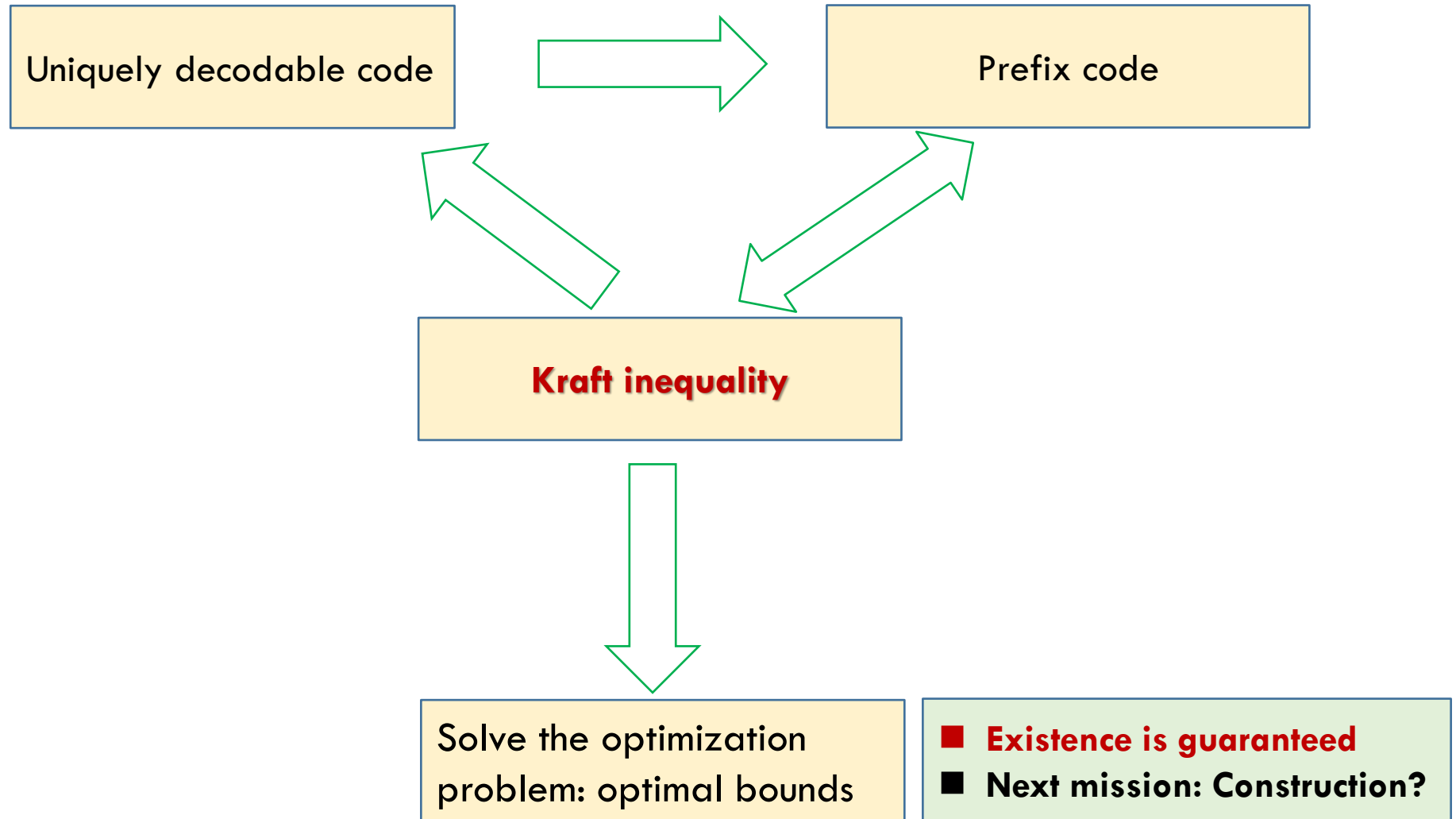
$$\leq \sum_{m=1}^{kl_{\max}} D^m D^{-m} = kl_{\max}$$

$$\sum_{x \in \mathcal{X}} D^{-l(x)} \leq (kl_{\max})^{\frac{1}{k}} \rightarrow 1, \text{ as } k \rightarrow \infty$$

l_{\max} is the maximum codeword length and $a(m)$ is the number of source sequences x^k mapping into codewords of length m .

Roadmap

Homework
Cover: 5.7, 5.8, 5.14, 5.16, 5.28



Summary

Related material: Cover, 5.1, 5.2, 5.3, 5.4, 5.5