

COMP2610 / 6261 — Information Theory

Lecture 13: Symbol Codes for Lossless Compression

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat  edu_assist_pro

17 September, 2018

Last time

Assignment Project Exam Help

Proof of the source coding theorem

- Fou
- Re

<https://eduassistpro.github.io>

The theor

- Could **variable length** coding help?
- Does entropy turn up for such codes as well?

Add WeChat edu_assist_pro

This time

Variable-length codes

Prefix codes

Kraft's inequality

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

- 1 Variable-Length Codes
 - Unique Decodability
 - Prefix Codes

Assignment Project Exam Help

- 2 The Kr

<https://eduassistpro.github.io>

- 3 Summary

Add WeChat edu_assist_pr

- 1 Variable-Length Codes
 - Unique Decodability
 - Prefix Codes

Assignment Project Exam Help

- 2 The kr

<https://eduassistpro.github.io>

- 3 Summary

Add WeChat edu_assist_pr

Codes: A Review

Notation:

- If \mathcal{A} is a finite set then \mathcal{A}^N is the set of all *strings of length N*.

- $\mathcal{A}^+ = \bigcup_{N \geq 1} \mathcal{A}^N$ is the set of all *finite strings*

Examples:

- $\{0, 1\}^3$

- $\{0, 1\}^+$

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Codes: A Review

Notation:

- If \mathcal{A} is a finite set then \mathcal{A}^N is the set of all *strings of length N*.

- $\mathcal{A}^+ = \bigcup_{N=1}^{\infty} \mathcal{A}^N$ is the set of all *finite strings*

Examples:

- $\{0, 1\}^3$
- $\{0, 1\}^+$

Binary S

Let X be an ensemble with $\mathcal{A}_X = \{a_1, \dots, a_n\}$.

A function $c : \mathcal{A}_X \rightarrow \{0, 1\}^+$ is a **code** for

- The binary string $c(x)$ is the **codeword**

Codes: A Review

Notation:

- If \mathcal{A} is a finite set then \mathcal{A}^N is the set of all *strings of length N*.

- $\mathcal{A}^+ = \bigcup_N \mathcal{A}^N$ is the set of all *finite strings*

Examples:

- $\{0, 1\}^3$
- $\{0, 1\}^+$

Binary S

Let X be an ensemble with $\mathcal{A}_X = \{a_1, \dots, a_l\}$.

A function $c : \mathcal{A}_X \rightarrow \{0, 1\}^+$ is a **code** for

- The binary string $c(x)$ is the **codeword**
- The **length** of the codeword for x is denoted $\ell(x)$.

Shorthand: $\ell_i = \ell(a_i)$ for $i = 1 \dots, l$.

Codes: A Review

Notation:

- If \mathcal{A} is a finite set then \mathcal{A}^N is the set of all *strings of length N* .

- $\mathcal{A}^+ = \bigcup_N \mathcal{A}^N$ is the set of all *finite strings*

Examples:

- $\{0, 1\}^3$
- $\{0, 1\}^+$

Binary S

Let X be an ensemble with $\mathcal{A}_X = \{a_1, \dots, a_l\}$.

A function $c : \mathcal{A}_X \rightarrow \{0, 1\}^+$ is a **code** for

- The binary string $c(x)$ is the **codeword**
- The **length** of the codeword for x is denoted $\ell(x)$.

Shorthand: $\ell_i = \ell(a_i)$ for $i = 1 \dots, l$.

- The **extension** of c assigns codewords to any sequence $x_1 x_2 \dots x_N$ from \mathcal{A}^+ by $c(x_1 \dots x_N) = c(x_1) \dots c(x_N)$

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let
- Sho

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let
- Sho
- All c

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let
- Sho
- All c
- The *extension* of c maps $\text{aba} \in \mathcal{A}_X^3$ \rightarrow

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let
- Sho
- All c
- The *extension* of c maps $aba \in \mathcal{A}_X^3$ $+$

Example 2 (Variable-Length Code):

Add WeChat edu_assist_pr

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let
- Sho
- All c
- The *extension* of c maps $\text{aba} \in \mathcal{A}_X^3$ \rightarrow

<https://eduassistpro.github.io>

Example 2 (Variable-Length Code):

- Let $c(a) = 0$, $c(b) = 10$, $c(c) = 110$, $c(d) = 111$

Add WeChat edu_assist_pr

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let
- Sho
- All c
- The *extension* of c maps $\text{aba} \in \mathcal{A}_X^3$ \rightarrow

<https://eduassistpro.github.io>

Example 2 (Variable-Length Code):

- Let $c(a) = 0$, $c(b) = 10$, $c(c) = 110$, $c(d) = 111$
- Shorthand: $C_2 = \{0, 10, 110, 111\}$

Add WeChat edu_assist_pr

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let
- Sho
- All c
- The *extension* of c maps $\text{aba} \in \mathcal{A}_X^3$ \rightarrow

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Example 2 (Variable-Length Code):

- Let $c(a) = 0$, $c(b) = 10$, $c(c) = 110$, $c(d) = 111$
- Shorthand: $C_2 = \{0, 10, 110, 111\}$
- In this case $\ell_1 = 1$, $\ell_2 = 2$, $\ell_3 = \ell_4 = 3$

Codes: A Review

Examples

X is an ensemble with $\mathcal{A}_X = \{a, b, c, d\}$

Assignment Project Exam Help

Example 1 (Uniform Code):

- Let
- Sho
- All c
- The *extension* of c maps $\text{aba} \in \mathcal{A}_X^3$ to 0100

<https://eduassistpro.github.io>

Example 2 (Variable-Length Code):

- Let $c(a) = 0$, $c(b) = 10$, $c(c) = 110$, $c(d) = 111$
- Shorthand: $C_2 = \{0, 10, 110, 111\}$
- In this case $\ell_1 = 1$, $\ell_2 = 2$, $\ell_3 = \ell_4 = 3$
- The *extension* of c maps $\text{aba} \in \mathcal{A}_X^3 \subset \mathcal{A}_X^+$ to 0100

Add WeChat edu_assist_pro

Unique Decodeability

Recall that a code is **lossless** if for all $x, y \in \mathcal{A}_X$

$$x \neq y \implies c(x) \neq c(y)$$

Assignment Project Exam Help

This ensures that if we work with a **single** outcome, we can uniquely decode the outcome

When we
require th

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Unique Decodeability

Recall that a code is **lossless** if for all $x, y \in \mathcal{A}_X$

$$x \neq y \implies c(x) \neq c(y)$$

This ensures that if we work with a **single** outcome, we can uniquely decode the outcome

When we
require th

Uniquely Decodable

A code c for X is **uniquely decodable** if no two different sequences of elements of X have the same codeword. That is, for all $\mathbf{x}, \mathbf{y} \in \mathcal{A}_X^+$

$$\mathbf{x} \neq \mathbf{y} \implies c(\mathbf{x}) \neq c(\mathbf{y})$$

This ensures that if we work with a **sequence** of outcomes, we can still uniquely decode the individual elements

Examples of uniquely decodable codes

Examples:

• $S_1 = \{0001, 0010, 0100, 1000\}$ is uniquely decodable

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Examples of uniquely decodable codes

Examples:

Assignment Project Exam Help

- $S_1 = \{0001, 0010, 0100, 1000\}$ is uniquely decodable
 - ▶ Uniform + Lossless \Rightarrow Uniquely decodable

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Examples of uniquely decodable codes

Examples:

- $C_1 = \{0001, 0010, 0100, 1000\}$ is uniquely decodable
 - ▶ Uniform + Lossless \Rightarrow Uniquely decodable

- C_2

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Examples of uniquely decodable codes

Examples:

- $C_1 = \{0001, 0010, 0100, 1000\}$ is uniquely decodable

- ▶ Uniform + Lossless \Rightarrow Uniquely decodable

- C_2

<https://eduassistpro.github.io>

- ▶ The code is of course lossless

Add WeChat edu_assist_pr

Examples of uniquely decodable codes

Examples:

- $C_1 = \{0001, 0010, 0100, 1000\}$ is uniquely decodable

- ▶ Uniform + Lossless \Rightarrow Uniquely decodable

- C_2

<https://eduassistpro.github.io>

- ▶ The code is of course lossless
- ▶ Lossless \Rightarrow Uniquely decodable

Add WeChat edu_assist_pro

Examples of uniquely decodable codes

Examples:

- $C_1 = \{0001, 0010, 0100, 1000\}$ is uniquely decodable
 - ▶ Uniform + Lossless \Rightarrow Uniquely decodable

- C_2

<https://eduassistpro.github.io>

- ▶ The code is of course lossless
- ▶ Lossless \nRightarrow Uniquely decodable

- $C_3 = \{0, 10, 110, 111\}$ is uniquely deco

Examples of uniquely decodable codes

Examples:

- $C_1 = \{0001, 0010, 0100, 1000\}$ is uniquely decodable

- ▶ Uniform + Lossless \Rightarrow Uniquely decodable

- C_2

<https://eduassistpro.github.io>

- ▶ The code is of course lossless
- ▶ Lossless \nRightarrow Uniquely decodable

- $C_3 = \{0, 10, 110, 111\}$ is uniquely deco

- ▶ We can easily segment a given code string scanning left to right

Examples of uniquely decodable codes

Examples:

- $C_1 = \{0001, 0010, 0100, 1000\}$ is uniquely decodable

- ▶ Uniform + Lossless \Rightarrow Uniquely decodable

- C_2

<https://eduassistpro.github.io>

- ▶ The code is of course lossless
- ▶ Lossless \nRightarrow Uniquely decodable

- $C_3 = \{0, 10, 110, 111\}$ is uniquely deco

- ▶ We can easily segment a given code string scanning left to right
- ▶ e.g. 0110010 \rightarrow 0, 110, 0, 10

“Self-punctuating” property

The code $C_3 = \{0, 10, 110, 111\}$ has a “self-punctuating” property

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

“Self-punctuating” property

The code $C_3 = \{0, 10, 110, 111\}$ has a “self-punctuating” property

Assignment Project Exam Help

Trivial to segment a given code string into individual codewords

- Ke
- On
stri

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

“Self-punctuating” property

The code $C_3 = \{0, 10, 110, 111\}$ has a “self-punctuating” property

Trivial to segment a given code string into individual codewords

- Ke
- On
stri

<https://eduassistpro.github.io>

Once our current segment matches a codeword, n

- Why? No codeword is a prefix of any other

Add WeChat [edu_assist_pro](#)

Not true for every uniquely decodable code, e.g. $C_4 = \{0, 01, 011\}$

- First bit 0 \rightarrow no certainty what the symbol is

Prefix Codes

a.k.a *prefix-free* or *instantaneous* codes

A simple property of codes **guarantees** unique decodeability

Prefix property

A codeword

+

rd

$c' \in \{0,$

$t.$

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Prefix Codes

a.k.a *prefix-free* or *instantaneous* codes

A simple property of codes **guarantees** unique decodeability

Prefix property

A codeword

+

rd

$c' \in \{0,$

t.

Can you cr

- **Example:** 01101 has prefixes 0, 01, 011, 011

Add WeChat edu_assist_pr

Prefix Codes

a.k.a *prefix-free* or *instantaneous* codes

A simple property of codes **guarantees** unique decodeability

Prefix property

A codeword

+

rd

$\mathbf{c}' \in \{0,$

t.

Can you cr

- **Example:** 01101 has prefixes 0, 01, 011, 011

Prefix Codes

A code $C = \{\mathbf{c}_1, \dots, \mathbf{c}_I\}$ is a **prefix code**

$i \in C$

there is no prefix of \mathbf{c}_i in C .

In a stream, no confusing one codeword with another

Prefix Codes: Examples

Examples:

- $C_1 = \{0001, 0010, 0100, 1000\}$ is prefix free

- $C_2 = \{1, 10, 110, 111\}$ is *not* prefix free

- $C'_2 = \{1, 10, 110, 111\}$ is *not* prefix free

- $C''_2 = \{1, 01, 110, 111\}$ is *not* prefix free since $c_3 = 110 = c_1 10$

Prefix Codes as Trees

$$C_1 = \{0001, 0010, 0100, 1000\}$$

	00	000	0000
			0001
		001	0010
		011	
	10	1	
1		1	1011
	11	110	1100
			1101
		111	1110
			1111

Prefix Codes as Trees

$$C_2 = \{0, 10, 110, 111\}$$

				0000
				0001
				0010
			001	
			011	
			1	
			1	
			110	1011
				1100
				1101
				1110
			111	1111

Prefix Codes as Trees

$$C'_2 = \{1, 10, 110, 111\}$$

			0000
		000	0001
	00	001	0010
			0111
		1	
	10	1	
1			1011
		110	1100
	11		1101
		111	1110
			1111

Each codeword choice eliminates its descendants

Prefix Codes are Uniquely Decodeable

Assignment Project Exam Help

- If $\ell^* = \max\{\ell_1, \dots, \ell_l\}$ then symbol is

$t \ell^*$

<https://eduassistpro.github.io>

$() = \dots$ $1 = a$

- ▶ If $\dots, d\}$
- ▶ If $\dots, d\}$
- ▶ If $\dots, d\}$

Add WeChat [edu_assist_pro](https://eduassistpro.github.io)

0	00	000	0000
			0001
			0010
1	10	100	1001
			1010
	101		1011
			1100
	11	110	1101
			1110
	111		1111

Uniquely Decodeable Codes are Not Always Prefix Codes

A uniquely decodeable code is **not necessarily** a prefix code

Example: $C_1 = \{0, 01, 011\}$

- 00..
- 010
- 011

Example: $C_2 = \{0, 01, 011, 111\}$

- This is the reverse of the prefix code

Relating various types of codes



Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Note that e.g.

Prefix \implies Uniquely Decodable

but

Not Prefix \nRightarrow Not Uniquely Decodable

Why prefix codes?

Assignment Project Exam Help

While prefix codes do not represent all uniquely decodable codes, they are convenient.

It will be easy

<https://eduassistpro.github.io>

Further, we can quickly establish if a given code is

- Testing for unique decodability is non-trivial.

Add WeChat edu_assist_pro

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$
- $L_2 = \{1, 2, 3, 3\}$
- $L_3 = \{2, 2, 3, 4, 4\}$
- $L_4 = \{1, 3, 3, 3, 3, 4\}$

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

0	01	010	
1	10	100	1001
			1010
		101	1011
			1100
	11	110	1101
			1110
		111	1111

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$ — $C_1 = \{0001, 0010, 0100, 1000\}$

- $L_2 = \{1, 2, 3, 3\}$

- $L_3 = \{2, 2, 3, 4, 4\}$

- $L_4 = \{1, 3, 3, 3, 3, 4\}$

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat [edu_assist_pro](#)

0			
		010	
	01		
		011	
1	10	100	
		101	1010
			1011
	11	110	1100
			1101
		111	1110
1111			

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$ — $C_1 = \{0001, 0010, 0100, 1000\}$

- $L_2 = \{1, 2, 3, 3\}$ — $C_2 = \{0, 10, 110, 111\}$

- $L_3 = \{2, 2, 3, 4\}$

- $L_4 = \{1, 3, 3, 3, 4\}$

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

0	01	010	0100
1	10	100	
		101	1010
			1011
	11	110	1100
			1101
		111	1110
			1111

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$ — $C_1 = \{0001, 0010, 0100, 1000\}$

- $L_2 = \{1, 2, 3, 3\}$ — $C_2 = \{0, 10, 110, 111\}$

- $L_3 = \{1, 2, 3, 4, 4\}$ — $C_3 = \{00, 01, 10, 110, 111\}$

- $L_4 = \{1, 3, 3, 3, 3, 4\}$

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

0		010	
1	10	100	1001
			1010
			1011
			1100
	11	110	1101
			1110
			1111

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$ — $C_1 = \{0001, 0010, 0100, 1000\}$

- $L_2 = \{1, 2, 3, 3\}$ — $C_2 = \{0, 10, 110, 111\}$

- $L_3 = \{2, 2, 3, 4, 4\}$ — $C_3 = \{00, 01, 10, 110, 111\}$

- $L_4 = \{1, 3, 3, 3, 3, 4\}$

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

0	01	10	
1	10	100	1001
			1010
		101	1011
			1100
	11	110	1101
			1110
		111	1111

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$ — $C_1 = \{0001, 0010, 0100, 1000\}$

- $L_2 = \{1, 2, 3, 3\}$ — $C_2 = \{0, 10, 110, 111\}$

- $L_3 = \{2, 2, 3, 4\}$ — $C_3 = \{00, 01, 100, 1000\}$

- $L_4 = \{1, 3, 3, 3, 3, 4\}$

<https://eduassistpro.github.io>

Add WeChat: edu_assist_pro

0			
	01		010
1	10	011	
		100	1001
			1010
			1011
	11	110	1100
			1101
			1110
		111	1111

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$ — $C_1 = \{0001, 0010, 0100, 1000\}$

- $L_2 = \{1, 2, 3, 3\}$ — $C_2 = \{0, 10, 110, 111\}$

- $L_3 = \{2, 2, 3, 4\}$ — $C_3 = \{00, 01, 100, 1010\}$

- $L_4 = \{1, 3, 3, 3, 3, 4\}$

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

0	01	100	1001
			1010
			1011
			1100
1	11	111	1101
			1110
			1111

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$ — $C_1 = \{0001, 0010, 0100, 1000\}$

- $L_2 = \{1, 2, 3, 3\}$ — $C_2 = \{0, 10, 110, 111\}$

- $L_3 = \{2, 2, 3, 4\}$ — $C_3 = \{00, 01, 100, 1010, 1011\}$

- $L_4 = \{1, 3, 3, 3, 3, 4\}$

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

0		010	
	10	100	
		101	1010
1	11	110	1101
		111	1110
			1111

Lengths and Trees

Suppose someone said “I want prefix codes with codewords lengths”:

- $L_1 = \{4, 4, 4, 4\}$ — $C_1 = \{0001, 0010, 0100, 1000\}$

- $L_2 = \{1, 2, 3, 3\}$ — $C_2 = \{0, 10, 110, 111\}$

- $L_3 = \{2, 2, 3, 4, 4\}$ — $C_3 = \{00, 01, 100, 1010, 1011\}$

- $L_4 = \{1, 3, 3, 3, 3, 4\}$ — Impossible!

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

0				
1	01		010	
			011	
	10		100	1001
			101	1010
				1011
	11		110	1100
			111	1101
				1110
			111	1111

The Kraft Inequality

a.k.a. The Kraft-McMillan Inequality

Kraft Inequality

For any prefix (binary) code C , its codeword lengths $\{\ell_1, \dots, \ell_I\}$ satisfy

$$\sum_{i=1}^I 2^{-\ell_i} \leq 1$$

Conversely, for any sequence $\{\ell_1, \dots, \ell_I\}$ of non-negative integers

there exists a prefix code C with those codeword lengths.

de

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

The Kraft Inequality

a.k.a. The Kraft-McMillan Inequality

Kraft Inequality

For any prefix (binary) code C , its codeword lengths $\{l_1, \dots, l_I\}$ satisfy

$$\sum_{i=1}^I 2^{-l_i} \leq 1$$

Conversely, for any sequence $\{l_1, \dots, l_I\}$ of non-negative integers, there exists a prefix code C with those codeword lengths.

de

Examples:

① $C_1 = \{0001, 0010, 0100, 1000\}$ is prefix and $\sum_{i=1}^4 2^{-4} = \frac{1}{4} \leq 1$

The Kraft Inequality

a.k.a. The Kraft-McMillan Inequality

Kraft Inequality

For any prefix (binary) code C , its codeword lengths $\{\ell_1, \dots, \ell_I\}$ satisfy

$$\sum_{i=1}^I 2^{-\ell_i} \leq 1$$

Conversely, for any sequence $\{\ell_1, \dots, \ell_I\}$

there exists a prefix code C with those codeword lengths.

de

Examples:

① $C_1 = \{0001, 0010, 0100, 1000\}$ is prefix and $\sum_{i=1}^4 2^{-\ell_i} = \frac{1}{4} \leq 1$

② $C_2 = \{0, 10, 110, 111\}$ is prefix and $\sum_{i=1}^4 2^{-\ell_i} = \frac{1}{2} + \frac{1}{4} + \frac{2}{8} = 1$

The Kraft Inequality

a.k.a. The Kraft-McMillan Inequality

Kraft Inequality

For any prefix (binary) code C , its codeword lengths $\{\ell_1, \dots, \ell_I\}$ satisfy

$$\sum_{i=1}^I 2^{-\ell_i} \leq 1$$

Conversely, for any sequence $\{\ell_1, \dots, \ell_I\}$ of non-negative integers, there exists a prefix code C with those codeword lengths.

de

Examples:

- 1 $C_1 = \{0001, 0010, 0100, 1000\}$ is prefix and $\sum_{i=1}^4 2^{-4} = \frac{1}{4} \leq 1$
- 2 $C_2 = \{0, 10, 110, 111\}$ is prefix and $\sum_{i=1}^4 2^{-\ell_i} = \frac{1}{2} + \frac{1}{4} + \frac{2}{8} = 1$
- 3 Lengths $\{1, 2, 2, 3\}$ give $\sum_{i=1}^4 2^{-\ell_i} = \frac{1}{2} + \frac{2}{4} + \frac{1}{8} > 1$ so no prefix code

Prefixes Exclude Codes

We are constrained when constructing prefix codes, as selecting a codeword eliminates a whole subtree

Choosing a prefix codeword of length 1 — e.g., $c(a) = 0$ — excludes:

Assignment Project Exam Help

● 2 x 2-bit codewords: 00, 01

0	00	000	0000
		001	0001
	01	0010	0010
		0011	0011
1	10		
	11	101	1011
		110	1100
		111	1110

<https://eduassistpro.github.io>

Add WeChat edu_assist_pro

Prefixes Exclude Codes

We are constrained when constructing prefix codes, as selecting a codeword eliminates a whole subtree

Choosing a prefix codeword of length 1 — e.g., $c(a) = 0$ — excludes:

Assignment Project Exam Help

0	00	000	0000
		001	0001
	01	010	0010
		011	0011
1	10	100	1000
		101	1001
		110	1010
		111	1011
	11	110	1100
		111	1101
		111	1110
		111	1111

• 2 x 2-bit codewords: 00, 01

• , 011}

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Prefixes Exclude Codes

We are constrained when constructing prefix codes, as selecting a codeword eliminates a whole subtree

Choosing a prefix codeword of length 1 — e.g., $c(a) = 0$ — excludes:

Assignment Project Exam Help

0	00	000	0000
		001	0001
		001	0010
	0		
1	1		
		101	1011
			1100
	11	110	1101
			1110
		111	1111
			1111

• 2 x 2-bit codewords: 00, 01

•

, 011}

0111}

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

Prefixes Exclude Codes

We are constrained when constructing prefix codes, as selecting a codeword eliminates a whole subtree

Choosing a prefix codeword of length 1 — e.g., $c(a) = 0$ — excludes:

Assignment Project Exam Help



• 2 x 2-bit codewords: 00, 01

•

•

•

$2^{k-\ell}$ x k -bit co

, 011}

0111}

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

For lengths $L = \{\ell_1, \dots, \ell_I\}$ and $\ell^* = \max\{\ell_1, \dots$

$$\sum_{i=1}^I 2^{\ell^* - \ell_i}$$

excluded ℓ^* -bit codewords.

Prefixes Exclude Codes

We are constrained when constructing prefix codes, as selecting a codeword eliminates a whole subtree

Choosing a prefix codeword of length 1 — e.g., $c(a) = 0$ — excludes:

Assignment Project Exam Help



• 2 x 2-bit codewords: 00, 01

•

•

•

•

$2^{k-\ell}$ x k -bit co

, 011}

0111}

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

For lengths $L = \{\ell_1, \dots, \ell_I\}$ and $\ell^* = \max\{\ell_1, \dots$

$$\sum_{i=1}^I 2^{\ell^* - \ell_i} \leq 2^{\ell^*}$$

excluded ℓ^* -bit codewords. But there are only 2^{ℓ^*} possible ℓ^* -bit codewords

Prefixes Exclude Codes

We are constrained when constructing prefix codes, as selecting a codeword eliminates a whole subtree

Choosing a prefix codeword of length 1 — e.g., $c(a) = 0$ — excludes:

Assignment Project Exam Help



• 2 x 2-bit codewords: 00, 01

•

•

•

•

$2^{k-\ell}$ x k -bit co

, 011}

0111}

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

For lengths $L = \{\ell_1, \dots, \ell_I\}$ and $\ell^* = \max\{\ell_1, \dots$

$$\frac{1}{2^{\ell^*}} \sum_{i=1}^I 2^{\ell^* - \ell_i} \leq 1$$

excluded ℓ^* -bit codewords. But there are only 2^{ℓ^*} possible ℓ^* -bit codewords

Prefixes Exclude Codes

We are constrained when constructing prefix codes, as selecting a codeword eliminates a whole subtree

Choosing a prefix codeword of length 1 — e.g., $c(a) = 0$ — excludes:

Assignment Project Exam Help



• 2 x 2-bit codewords: 00, 01

•

•

•

•

$2^{k-\ell}$ x k -bit co

, 011}

0111}

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr

For lengths $L = \{\ell_1, \dots, \ell_I\}$ and $\ell^* = \max\{\ell_1, \dots$

$$\sum_{i=1}^I 2^{-\ell_i} \leq 1$$

excluded ℓ^* -bit codewords. But there are only 2^{ℓ^*} possible ℓ^* -bit codewords

Kraft inequality: other direction

Suppose we are given lengths satisfying

$$\sum_{i=1}^l 2^{-l_i} \leq 1$$

Assignment Project Exam Help

Then, we can

- Pick a node at depth l_i and assign it the first codeword
- Removing all descendants of the node (to ensure the prefix condition)
- Picking the next (remaining) node at depth l_j and assigning it the second codeword
- Removing all descendants of the node (to ensure the prefix condition)
- \vdots

Kraft inequality: comments

Kraft's inequality actually holds more generally for uniquely decodable codes

- Harder to prove

Note that if

$$\sum_{i=1}^I 2^{-\ell_i} \leq 1$$

it does not mean the **given** code necessarily

- Just that we can **construct** a prefix code with these lengths

Summary

Key ideas from this lecture:

- **Prefix** and **Uniquely Decodable** variable-length codes

- Prefix codes are tree-like

- Eve

- The



$$\sum_i 2^{-\ell_i} \leq 1$$

exists

- ▶ Prefix/U.D. code implies $\sum_i 2^{-\ell_i} \leq 1$

Relevant Reading Material:

- MacKay: §5.1 and §5.2
- Cover & Thomas: §5.1, §5.2, and §5.5

Next time

Bound on expected length for a prefix code

Shannon codes

Huffman coding

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu_assist_pr