

Channel Radix $r < q$

Symbols are sent from the source through a channel.

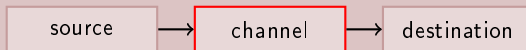


Figure: Communication System

- If the channel can carry q or more different types of values, sending a message s through the channel is trivial².
- If not, we need to *encode* s more cleverly to send it through the channel.

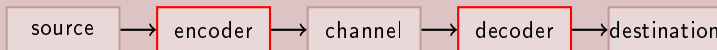


Figure: Communication System with Encoder & Decoder

We will thus assume $r < q$, and construct codes which utilise all r possible values. We call these r -ary codes. Usually, $r = 2$ (binary codes).

²Symbol-to-symbol mapping

Definition of Unique Decodability

Definition (Unique Decodability)

A code \mathcal{C} is *uniquely decodable* iff $\mathcal{C}^* : S^* \rightarrow T^*$ is a injection.

We say that codes that are not uniquely decodable are *ambiguous*.

Example (\mathcal{S} : Unbiased Die, \mathcal{C} : Binary Code)

Since $\mathcal{C}^*(3) = \mathcal{C}^*(11) = 11$, \mathcal{C} is ambiguous.

Lecture 1: Unique Decodability

Source

- Preliminaries
- Probability Theory
- Notation
- Definition
- Assumptions
- Examples

Source Code

- Motivation
- Definition
- Examples

Unique Decodability

- Definition**
- Block Codes
- Good Codes
- Sardinas-Patterson

Epilogue

- Summary
- Next Session

Examples: The Biased Die, binary code

Assume index-mapping of source symbols to probabilities & code words.

Example

Source could produce the symbols (6, 5, 4, 3, 2, 1) w. probability distribution $(\frac{1}{2}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10})$.

- 1) Mapping to binary representation: *ambiguous* (ex: 21 = 5).
 • $\mathcal{C} = (110, 101, 100, 11, 10, 1)$
- 2) Block code: *sub-optimal*. $L(\mathcal{C}) = 3$.
 • $\mathcal{C} = (110, 101, 100, 011, 010, 001)$
- 3) Clever encoding; $L(\mathcal{C}) = 2 + \frac{2}{5}$: *unbounded decoding delay*.
 • $\mathcal{C} = (0, 01, 011, 0111, 01111, 11111)$
- 4) The fix; $L(\mathcal{C}) = 2 + \frac{3}{5}$: *bounded decoding delay: 1*.
 • $\mathcal{C} = (0, 01, 011, 0111, 01111, \underline{0}11111)$

Lecture 2: Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

What we have seen

- Source coding is not trivial.

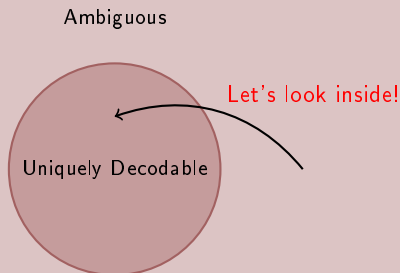


Figure: Classes of Source Codes.

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Definition of Instantaneous Codes

Definition (Instantaneous Code)

A code \mathcal{C} is *instantaneous* if any code word sequence can be decoded “on the fly”. That is,

$$\forall \mathbf{w}; \mathbf{s} \mapsto \mathbf{w} \text{ . } \nexists \mathbf{t} \in T^+, \mathbf{s}' ; \mathbf{s}' \mapsto \mathbf{w}\mathbf{t} \text{ . } \mathbf{s} \text{ does not prefix } \mathbf{s}' \text{ .}$$

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code**Definition**Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Definition of Prefix Codes

Definition (Prefix Code)

A code \mathcal{C} is a *prefix* code if no code-word w_i is a prefix of any other code-word w_j , with $i \neq j$. That is,

$$\forall w_i, w_j \ . \ \nexists t \in T^+ \ . \ w_j = w_i t$$

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code**Definition**Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Instantaneous codes are Prefix codes, and vice versa

Theorem (Instantaneous iff Prefix)

A code \mathcal{C} is instantaneous iff it is a prefix code.

Proof.

We prove each direction of the iff.

$\neg \Leftarrow \neg$: Assume \mathcal{C} is not a prefix code. Let w_i be a prefix of w_j (these exist by definition).

Lecture 2:
Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

**Instantaneous iff
Prefix**

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Idea

Recall the reversed code

$$\mathcal{C} = (0, 10, 110, 1110, 11110, 11111).$$

Each word w_i can be considered an element in the following T^* tree.

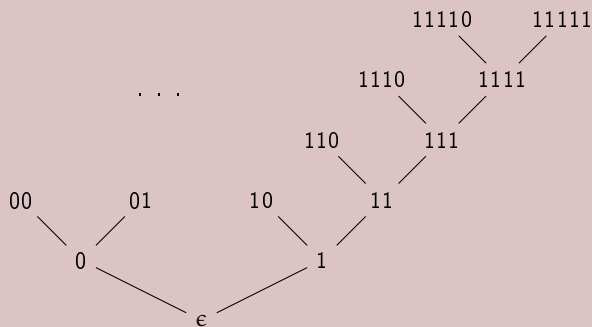


Figure: Prefix Code Construction, binary

$$\mathcal{C} = \{ \quad \quad \quad \}$$

Lecture 2:
Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

$$\mathcal{C} = (0, 10, 110, 1110, 11110, 11111).$$

Each word w_i can be considered an element in the following T^* tree.

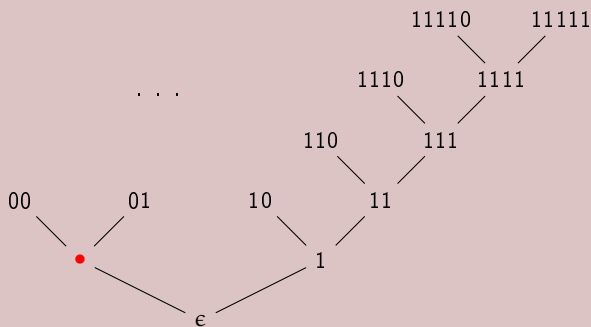


Figure: Prefix Code Construction, binary

$$\mathcal{C} = \{0, \dots, \ell-1\}$$

Idea

Recall the reversed code

$$\mathcal{C} = (0, 10, 110, 1110, 11110, 11111).$$

Each word w_i can be considered an element in the following T^* tree.

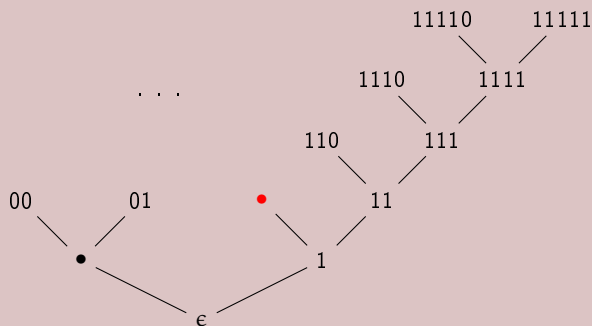


Figure: Prefix Code Construction, binary

$$\mathcal{C} = \{0, 10, \quad \quad \quad \}$$

Lecture 2:
Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Idea

Recall the reversed code

$$\mathcal{C} = (0, 10, 110, 1110, 11110, 11111).$$

Each word w_i can be considered an element in the following T^* tree.

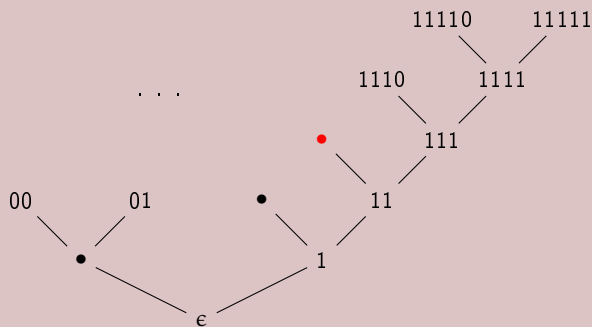


Figure: Prefix Code Construction, binary

$$\mathcal{C} = \{0, 10, 110, \quad \quad \quad \}$$

Lecture 2:
Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Idea

Recall the reversed code

$$\mathcal{C} = (0, 10, 110, 1110, 11110, 11111).$$

Each word w_i can be considered an element in the following T^* tree.

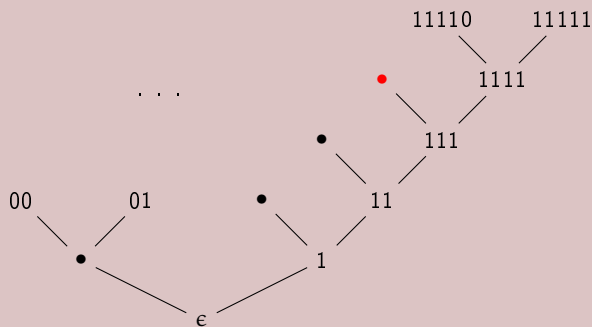


Figure: Prefix Code Construction, binary

$$\mathcal{C} = \{0, 10, 110, 1110, \quad \quad \quad \}$$

Lecture 2:
Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Idea

Recall the reversed code

$$\mathcal{C} = (0, 10, 110, 1110, 11110, 11111).$$

Each word w_i can be considered an element in the following T^* tree.

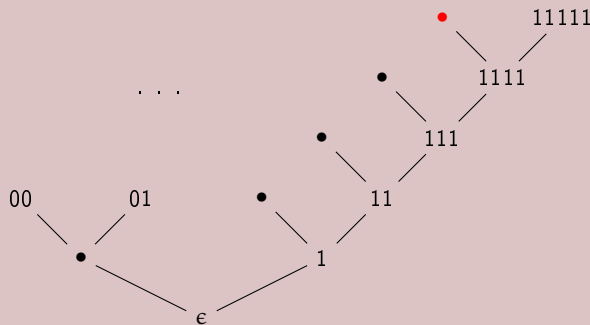


Figure: Prefix Code Construction, binary

$$\mathcal{C} = \{0, 10, 110, 1110, 11110, \quad \}$$

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Idea

Recall the reversed code

$$\mathcal{C} = (0, 10, 110, 1110, 11110, 11111).$$

Each word w_i can be considered an element in the following T^* tree.

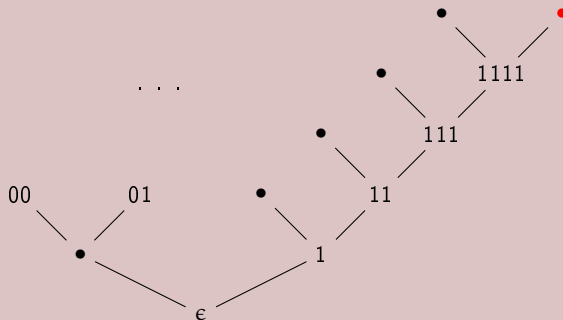


Figure: Prefix Code Construction, binary

$$\mathcal{C} = \{0, 10, 110, 1110, 11110, 11111\}$$

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Designing Codes By Word Length

Note that

- Each parent is a prefix of its children.
- Picking a word as a codeword will thus exclude all its subtrees¹.
- So, choosing 0 to be in \mathcal{C} will reserve $\frac{1}{2} = \frac{1}{2^1} = \frac{1}{2^{|0|}}$ the tree².
- Choosing 01 to be in \mathcal{C} will reserve $\frac{1}{4} = \frac{1}{2^2} = \frac{1}{2^{|01|}}$ the tree.
- Choosing w , where $|w| = l$, to be in \mathcal{C} will reserve $\frac{1}{2^l}$ the tree.

Note also that this approach, and the above observation, is generalisable to trees of *any* arity r ; replace $\frac{1}{2^l}$ with $\frac{1}{r^l}$.

Example

Is it possible to design a *ternary* code with word lengths 1, 2, 3, 3, 4?

¹else we violate the prefix property

²**Note:** You can also consider this a fraction of *the leaves of the tree*.

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Designing Codes By Word Length

Note that

- Each parent is a prefix of its children.
- Picking a word as a codeword will thus exclude all its subtrees¹.
- So, choosing 0 to be in \mathcal{C} will reserve $\frac{1}{2} = \frac{1}{2^1} = \frac{1}{2^{|0|}}$ the tree².
- Choosing 01 to be in \mathcal{C} will reserve $\frac{1}{4} = \frac{1}{2^2} = \frac{1}{2^{|01|}}$ the tree.
- Choosing w , where $|w| = l$, to be in \mathcal{C} will reserve $\frac{1}{2^l}$ the tree.

Note also that this approach, and the above observation, is generalisable to trees of *any* arity r ; replace $\frac{1}{2^l}$ with $\frac{1}{r^l}$.

Example

Is it possible to design a *ternary* code with word lengths 1, 2, 3, 3, 4? Yes.

$$\frac{1}{3} + \frac{1}{3^2} + \frac{1}{3^3} + \frac{1}{3^3} + \frac{1}{3^4} = \frac{43}{81} \leq 1.$$

How about binary?

¹else we violate the prefix property

²**Note:** You can also consider this a fraction of *the leaves of the tree*.

Lecture 2: Optimal Codes

Recap

Definitions
Motivation

Instantaneous
Code

Definition
Instantaneous iff
Prefix

Construction

Inequalities

Kraft
McMillan
Sum Up

Optimal Codes

Do We Stop
Here?
Definition and
Existence
Huffman
Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Designing Codes By Word Length

Note that

- Each parent is a prefix of its children.
- Picking a word as a codeword will thus exclude all its subtrees¹.
- So, choosing 0 to be in \mathcal{C} will reserve $\frac{1}{2} = \frac{1}{2^1} = \frac{1}{2^{|0|}}$ the tree².
- Choosing 01 to be in \mathcal{C} will reserve $\frac{1}{4} = \frac{1}{2^2} = \frac{1}{2^{|01|}}$ the tree.
- Choosing w , where $|w| = l$, to be in \mathcal{C} will reserve $\frac{1}{2^l}$ the tree.

Note also that this approach, and the above observation, is generalisable to trees of *any* arity r ; replace $\frac{1}{2^l}$ with $\frac{1}{r^l}$.

Example

Is it possible to design a *ternary* code with word lengths 1, 2, 3, 3, 4? Yes.

$$\frac{1}{3} + \frac{1}{3^2} + \frac{1}{3^3} + \frac{1}{3^3} + \frac{1}{3^4} = \frac{43}{81} \leq 1.$$

How about binary? No.

$$\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^4} = 1 + \frac{1}{16} > 1.$$

¹else we violate the prefix property

²**Note:** You can also consider this a fraction of *the leaves* of the tree.

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Kraft's Inequality

Kraft's inequality should now come as no surprise.

Theorem (Kraft's Inequality)

There is an instantaneous r -ary code \mathcal{C} with word-lengths l_1, \dots, l_q iff

$$\sum_{i=1}^q \frac{1}{r^{l_i}} \leq 1. \quad (1)$$

Proof

\Leftarrow : ^a Renumber l_i s.t. $l_1 \leq \dots \leq l_q$. Let $l = l_q$. ^b Consider the tree $T^{\leq l} = T^0 \cup \dots \cup T^l$, of height l . It is finite. It has r^h vertices at each height $0 \leq h \leq l$. Specifically, it has r^l leaves; r^{l-h} leaves above a word in height h .

Cont.

^a**Idea:** Show that due to (1), we can pick words in the tree w/o pruning all leaves.

^bMaximum length

Lecture 2: Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

Proof

Start w. l_1 : Pick a word w_1 at height l_1 from $T^{\leq l}$, prune it, and its subtrees. This prunes r^{l-l_1} leaves. Now note that (if $q > 1$)

$$r^{l-l_1} = r^l * \frac{1}{r^{l_1}} < r^l \sum_{i=1}^q \frac{1}{r^{l_i}} \leq r^l,$$

last inequality by (1)^a Due to the “<”, there must be leaves left in $T^{\leq l}$. Thus there are words remaining at *any* desired height l_i .

Proceed: Picking w_k at height l_k prunes r^{l-l_k} leaves for $k < q$, yielding

$$r^l \sum_{i=1}^k \frac{1}{r^{l_i}} < r^l \sum_{i=1}^q \frac{1}{r^{l_i}} \leq r^l.$$

Eventually you reach $k = q - 1$. Only l_q , the largest word length, remains. Due to “<”, there are still leaves left. Pick a leaf for w_q ; these have length l_q . You are done.

The process ensures that no w_j prefixes a $w_{j'}$. So \mathcal{C} is a prefix code, and thus instantaneous.

Cont.

^aMultiply both sides with r^l .

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Proof.

\Rightarrow : Since \mathcal{C} is instantaneous, it is a prefix code. Thus each leaf of $T^{\leq l}$ is above at most 1 code-word of \mathcal{C} . Each $w_i \in \mathcal{C}$ is below r^{l-l_i} leaves. There are thus $\sum_i r^{l-l_i}$ leaves above code words. Finally,

$$\sum_{i=1}^q r^{l-l_i} = r^l \sum_{i=1}^q \frac{1}{r^{l_i}} \leq r^l,$$

where the last inequality must hold since we never count leaves twice^a. (1) follows by division w. r^l .



^aLeaves above code-words must thus be fewer than all leaves

Lecture 2:
Optimal Codes

Recap

- Definitions
- Motivation

Instantaneous
Code

- Definition
- Instantaneous iff Prefix
- Construction

Inequalities

Kraft

- McMillan
- Sum Up

Optimal Codes

- Do We Stop Here?

- Definition and Existence

- Huffman

- Huffman codes are Instantaneous (binary)

Epilogue

- The Best Source Codes

McMillan's Inequality

Strangely enough, the condition is **not** weaker if we “relax” our requirement to just any uniquely decodable codes.

Theorem (McMillan's Inequality)

There is a uniquely decodable r -ary code \mathcal{C} with word-lengths l_1, \dots, l_q iff

$$\sum_{i=1}^q \frac{1}{r^{l_i}} \leq 1. \quad (2)$$

Proof

\Leftarrow : Follows from (1) (Kraft).

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

McMillan's Inequality

Strangely enough, the condition is **not** weaker if we “relax” our requirement to just any uniquely decodable codes.

Theorem (McMillan's Inequality)

There is a uniquely decodable r -ary code \mathcal{C} with word-lengths l_1, \dots, l_q iff

$$\sum_{i=1}^q \frac{1}{r^{l_i}} \leq 1. \quad (2)$$

Proof

\Leftarrow : Follows from (1) (Kraft).

\Rightarrow : ^a Let \mathcal{C} be a uniquely decodable r -ary code with word-lengths l_1, \dots, l_q . Let $l = \max(l_1, \dots, l_q)$, $m = \min(l_1, \dots, l_q)$, and

$$K = \sum_{i=1}^q \frac{1}{r^{l_i}}.$$

Cont.

^aAwesome arithmetic ensues. Suggestion: Follow the steps w. paper & pencil.

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Proof

Consider^a, for $n \geq 1$,

$$K^n = \left(\sum_{i=1}^q \frac{1}{r^{l_i}} \right)^n.$$

K^n is a sum of terms

$$\frac{1}{r^{l_{i_1}}} * \frac{1}{r^{l_{i_2}}} * \cdots * \frac{1}{r^{l_{i_n}}} = \frac{1}{r^j}; \quad j = l_{i_1} + \cdots + l_{i_n}. \quad (3)$$

So, j is always a sum of some n word-lengths. Thus, $mn \leq j \leq ln$.
Gathering this, we get

$$K^n = \sum_{j=mn}^{ln} \frac{N_{j,n}}{r^j}.$$

Check the index j . $N_{j,n}$ is the nr. of ways to write j on the form in (3). Now, j is a *sum of word-lengths*. As such, $N_{j,n}$ is the nr. of sequences w_{i_1}, \dots, w_{i_n} of length j , which forms $\mathbf{w} = w_{i_1} \cdots w_{i_n}$.

Cont.

^aReally *consider* it. Try an example w. $n = 2$, for instance.

Lecture 2:
Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

Proof.

\mathcal{C} is uniquely decodable. Thus each w arises from at most one such sequence w_{i_1}, \dots, w_{i_n} of length j . There are at most r^j such sequences^a, so $N_{j,n} \leq r^j$. So

$$K^n = \sum_{j=mn}^{\ln} \frac{N_{j,n}}{r^j}.$$

consists of $\ln + 1 - mn$ terms on form $\frac{N_{j,n}}{r^j} \leq 1$. Thus,

$$K^n \leq (l - m)n + 1.$$

Finally^b, assume $K > 1$. However, in the above, K^n grows exponentially and $(l - m)n + 1$ linearly for $n \rightarrow \infty$. A contradiction. Thus, $K \leq 1$ must hold.



^aSequences have length r ; $|T^j| = r^j$.

^bHere comes the amazement.

Lecture 2:
Optimal Codes

Recap

- Definitions
- Motivation

Instantaneous
Code

- Definition
- Instantaneous iff Prefix
- Construction

Inequalities

- Kraft
- McMillan**
- Sum Up

Optimal Codes

- Do We Stop Here?
- Definition and Existence
- Huffman
- Huffman codes are Instantaneous (binary)

Epilogue

- The Best Source Codes

The Surprise

These results put together:

Corollary

There is an instantaneous r -ary code with word-lengths l_1, \dots, l_q iff there is a uniquely decodable r -ary code with these word-lengths.

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Is Our Instantaneous Code Creation Scheme Good Enough?

It gets better!

Example (Best binary code for the biased die)

Recall our clever encoding

$$\mathcal{C} = (0, 01, 011, 0111, 01111, 11111)$$

It has average word length $L(\mathcal{C}) = 2 + \frac{2}{5}$.

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

**Do We Stop
Here?**

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Is Our Instantaneous Code Creation Scheme Good Enough?

It gets better!

Example (Best binary code for the biased die)

Recall our clever encoding

$$\mathcal{C} = (0, 01, 011, 0111, 01111, 11111)$$

It has average word length $L(\mathcal{C}) = 2 + \frac{2}{5}$. It is not clever enough, since this one is better.

$$\mathcal{C} = (0, 101, 1000, 1001, 110, 111).$$

Here, $L(\mathcal{C}) = \frac{1}{2} * 1 + \frac{1}{10}(3 + 4 + 4 + 3 + 3) = \frac{22}{10} = 2 + \frac{1}{5}$; *an improvement!*

Punchline: This “best code” was produced by an *algorithm*!

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff

Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

**Do We Stop
Here?**

Definition and
Existence

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

So, there are *more interesting* instantaneous codes.

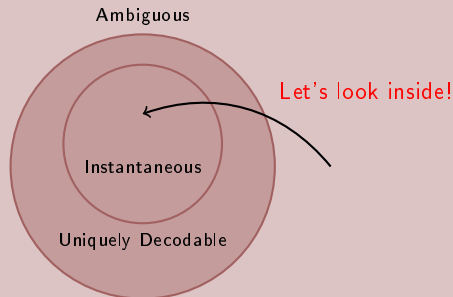


Figure: Classes of Source Codes.

We want to have:

- \mathcal{C} *instantaneous*, and to
- *minimise* $L(\mathcal{C})$,

Lecture 2: Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

Consider only Instantaneous Codes

By Kraft and McMillan, we have

Lemma

For a given source S , the set L of all average word-lengths $L(\mathcal{C})$ of uniquely decodable r -ary codes \mathcal{C} for S is equal to that set for instantaneous r -ary codes \mathcal{C} for S .

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff

Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

**Definition and
Existence**

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Consider only Instantaneous Codes

By Kraft and McMillan, we have

Lemma

For a given source S , the set L of all average word-lengths $L(\mathcal{C})$ of uniquely decodable r -ary codes \mathcal{C} for S is equal to that set for instantaneous r -ary codes \mathcal{C} for S .

Definition (Optimal Code)

An instantaneous code \mathcal{D} is an *optimal code* for a source S if for any other instantenous code \mathcal{C} , we have $L(\mathcal{D}) \leq L(\mathcal{C})$.

Is there such a \mathcal{D} ? L is bounded below (by 0). Let $L_{\min}(S)$ be the glb of L . Is it $L_{\min}(S) = 0$? If so, we can *keep improving our code* to approach $L_{\min}(S)$.

Lecture 2: Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

At some point, we hit rock-bottom.

Theorem (Existence of Optimal Codes)

Each source \mathcal{S} has an optimal r -ary code.

Proof.

There exists an instantaneous r -ary code \mathcal{C} for \mathcal{S} ; block code $l_1 = \dots = l_q = l$ s.t. $r^l \geq q$ (leaves of the tree). \mathcal{C} exists by Kraft.

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

**Definition and
Existence**

Huffman
Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

At some point, we hit rock-bottom.

Theorem (Existence of Optimal Codes)

Each source \mathcal{S} has an optimal r -ary code.

Proof.

There exists an instantaneous r -ary code \mathcal{C} for \mathcal{S} ; block code $l_1 = \dots = l_q = l$ s.t. $r^l \geq q$ (leaves of the tree). \mathcal{C} exists by Kraft. We now show there are only finitely many \mathcal{D} w. $L(\mathcal{D}) \leq L(\mathcal{C})$ (better codes)^a. There must then be a least $L(\mathcal{D})$.

Renumber \mathcal{S}, \mathcal{P} s.t. $p_i > 0$ for $i \leq k$ and $p_i = 0$ for $i > k$. Let $p = \min(p_1, \dots, p_k) > 0$. Let $L(\mathcal{D}) \leq L(\mathcal{C})$. For word-lengths of \mathcal{D} , we have

$$l_i \leq \frac{L(\mathcal{C})}{p}; \quad i = 1, \dots, \underline{k},$$

for else $L(\mathcal{D}) = p_1 l_1 + \dots + p_q l_q \geq p_i l_i > p \frac{L(\mathcal{C})}{p} = L(\mathcal{C})$, violating definition of \mathcal{D} . Finite $w \in T^+$ s.t. $|w| \leq \frac{L(\mathcal{C})}{p} \implies$ finite choices for w_1, \dots, w_k in \mathcal{D}

There are infinite choices for $i > k$, but these word nay contribute to $L(\mathcal{D})$. Result follows. □

^a L forms a finite *total order*, bounded below

Lecture 2: Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff

Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?

**Definition and
Existence**

Huffman

Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Huffman Coding

The idea: “flatten” P s.t. the code, if considered a source, outputs symbols w_i a *uniform distribution*.

Algorithm 1: Construction of a r -ary Huffman code

Input: A probability distribution $P = (p_1, \dots, p_n)$

Result: An association of code words $\mathcal{C} = (w_1, \dots, w_n)$

```

1 if  $n \leq r$  then
2   | Let  $\mathcal{C} = (0, 1, \dots, n-1)$ 
3 else
4   | Renumerate  $P$  if necessary s.t.  $p_1 \geq p_2 \geq \dots \geq p_n$ 
5   | Compute  $s$  s.t.  $2 \leq s \leq r$  and  $s \equiv n \pmod{r-1}$ 
6   | Let  $P' = (p_1, \dots, p_{n-s}, p')$ , where  $p' = p_{n-s+1} + \dots + p_n$ 
7   | Execute the algorithm recursively on  $P'$  with result
      |  $\mathcal{C}' = (w_1, \dots, w_{n-s}, t)$ 
8   | Let  $\mathcal{C} = (w_1, \dots, w_{n-s}, t0, t1, \dots, t(s-1))$ 
9 return  $\mathcal{C}$ 

```

Lecture 2: Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

Huffman Example

0.5

0.1

0.1

0.1

0.1

0.1

Figure: Huffman Code Generation for Biased Die

Lecture 2:
Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence**Huffman**Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Huffman Example

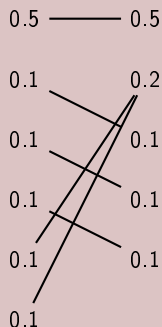


Figure: Huffman Code Generation for Biased Die

Lecture 2:
Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence**Huffman**Huffman codes
are Instantaneous
(binary)

Epilogue

The Best Source
Codes

Huffman Example

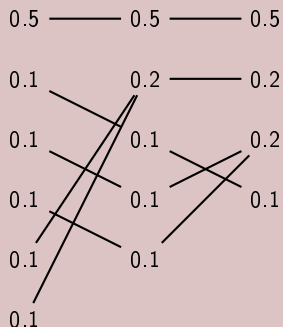


Figure: Huffman Code Generation for Biased Die

Lecture 2:
Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

Huffman Example

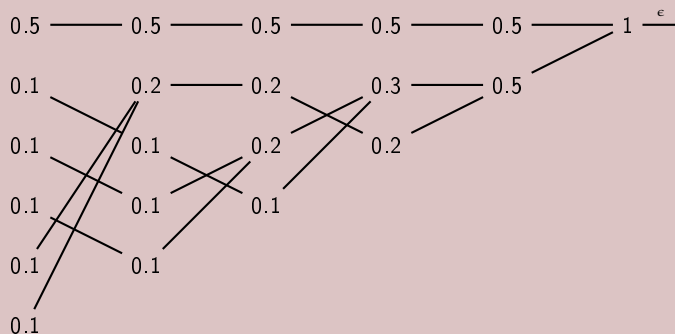


Figure: Huffman Code Generation for Biased Die

Lecture 2:
Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

Huffman Example

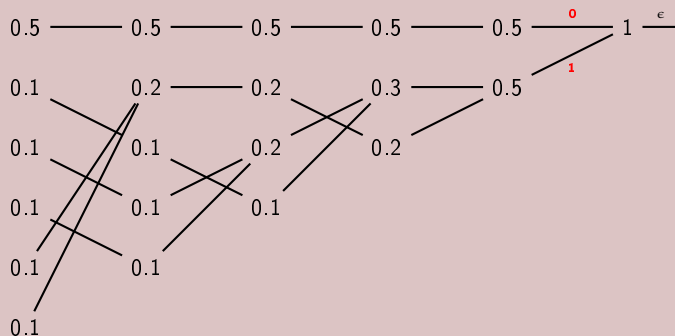


Figure: Huffman Code Generation for Biased Die

Lecture 2:
Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

Huffman Example

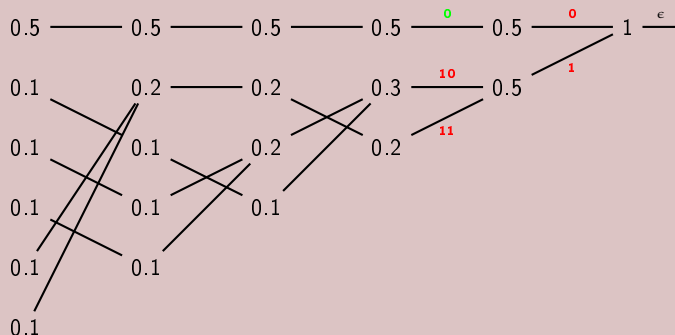


Figure: Huffman Code Generation for Biased Die

Lecture 2:
Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

Huffman Example

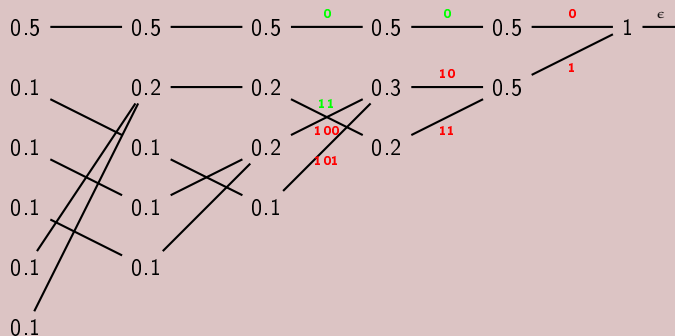


Figure: Huffman Code Generation for Biased Die

Lecture 2: Optimal Codes

Re cap

Definitions

Motivation

Inequalities

Kraft

McMillan

Optimal Codes

Huffman

Epilogue

Huffman Example

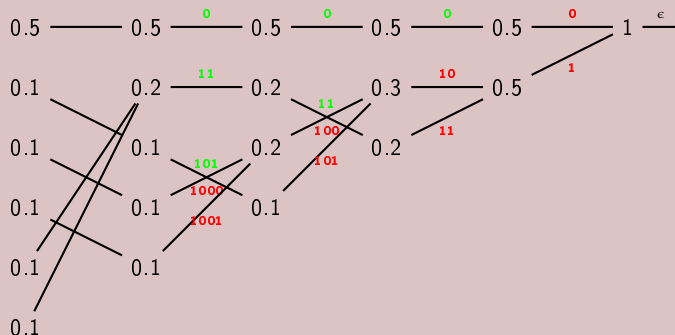


Figure: Huffman Code Generation for Biased Die

Lecture 2:
Optimal Codes

Recap
Definitions
Motivation
Instantaneous Code
Definition
Instantaneous iff Prefix
Construction
Inequalities
Kraft
McMillan
Sum Up
Optimal Codes
Do We Stop Here?
Definition and Existence
Huffman
Huffman codes are Instantaneous (binary)
Epilogue
The Best Source Codes

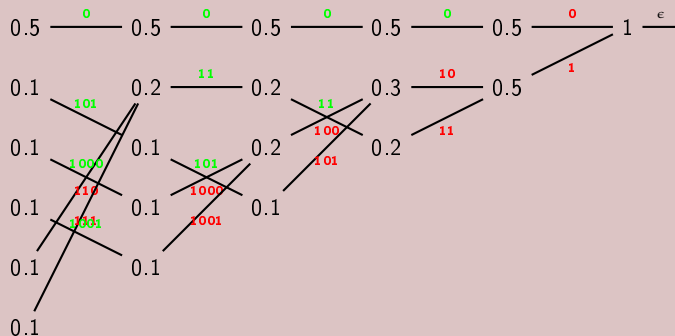


Figure: Huffman Code Generation for Biased Die

Resulting code: $\mathcal{C} = (0, 101, 1000, 1001, 110, 111)$.

Lecture 2: Optimal Codes

Re cap

Definitions

Motivation

Inequalities

Kraft

McMillan

Optimal Codes

Huffman

Epilogue

Huffman Codes are Instantaneous

Let

$$\mathcal{S} : s_1, \dots, s_{q-2}, \underbrace{s_{q-1}, s_q}$$

$$\mathcal{S}' : s_1, \dots, s_{q-2}, \quad s'$$

$$\mathcal{P} : p_1, \dots, p_{q-2}, \underbrace{p_{q-1}, p_q}$$

$$\mathcal{P}' : p_1, \dots, p_{q-2}, \quad p'$$

$$\mathcal{C} : w_1, \dots, w_{q-2}, \underbrace{w'0, w'1}$$

$$\mathcal{C}' : w_1, \dots, w_{q-2}, \quad w'$$

Lemma

If \mathcal{C}' is instantaneous, so is \mathcal{C} .

Proof.

Prefix tree. □Lecture 2:
Optimal Codes

Recap

Definitions

Motivation

Instantaneous
Code

Definition

Instantaneous iff
Prefix

Construction

Inequalities

Kraft

McMillan

Sum Up

Optimal Codes

Do We Stop
Here?Definition and
Existence

Huffman

**Huffman codes
are Instantaneous
(binary)**

Epilogue

The Best Source
Codes

Optimal Codes

We have shown the existence of *optimal* source codes, and shown how to *construct* them.

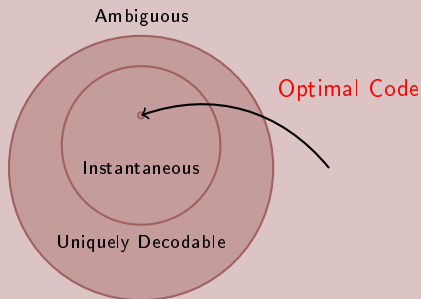


Figure: Classes of Source Codes.

We will finish off source coding in the next session by **proving** Huffman codes are optimal, and discuss them further.

Lecture 2: Optimal Codes

- Recap
- Definitions
- Motivation
- Instantaneous Code
 - Definition
 - Instantaneous iff Prefix
 - Construction
- Inequalities
 - Kraft
 - McMillan
 - Sum Up
- Optimal Codes
 - Do We Stop Here?
 - Definition and Existence
 - Huffman
 - Huffman codes are Instantaneous (binary)
- Epilogue
- The Best Source Codes

An Interesting Note

How much does $L(\mathcal{C})$ differ from $L(\mathcal{C}')$?

$$\begin{aligned}
 L(\mathcal{C}) - L(\mathcal{C}') &= (p_{q-1} + p_q)(l+1) - (p_{q-1} + p_q)l \\
 &= p_{q-1} + p_q \\
 &= p'.
 \end{aligned}
 \tag{\mathcal{C}-\mathcal{C}'\text{-Ldiff}}$$

Lecture 3: A Measure of Information

Optimality of Huffman Codes

$L(\mathcal{C})$ of Huffman Codes

Huffman Codes are Optimal

Source Extension

Information

$I(s)$ measure

$I(s)$ is uniquely defined

Entropy

$H(S)$ measure

Properties of $H(S)$

Epilogue

An Interesting Note

How much does $L(\mathcal{C})$ differ from $L(\mathcal{C}')$?

$$\begin{aligned}
 L(\mathcal{C}) - L(\mathcal{C}') &= (p_{q-1} + p_q)(l+1) - (p_{q-1} + p_q)l \\
 &= p_{q-1} + p_q \\
 &= p'.
 \end{aligned}
 \tag{\mathcal{C}-\mathcal{C}'\text{-Ldiff}}$$

Recall that $L(\mathcal{C}^{q-1}) = 1 * |\epsilon| = 0$. Iterating (p. 26):

$$L(\mathcal{C}) = p' + p'' + \dots + p^{(q-1)}$$

A convenient way to find $L(\mathcal{C})$ *without constructing* \mathcal{C} .

An Interesting Note

How much does $L(\mathcal{C})$ differ from $L(\mathcal{C}')$?

$$\begin{aligned} L(\mathcal{C}) - L(\mathcal{C}') &= (p_{q-1} + p_q)(l+1) - (p_{q-1} + p_q)l \\ &= p_{q-1} + p_q \\ &= p'. \end{aligned} \quad (\mathcal{C}-\mathcal{C}'\text{-Ldiff})$$

Recall that $L(\mathcal{C}^{q-1}) = 1 * |\epsilon| = 0$. Iterating (p. 26):

$$L(\mathcal{C}) = p' + p'' + \dots + p^{(q-1)}$$

A convenient way to find $L(\mathcal{C})$ *without constructing* \mathcal{C} .

Example (Biased die, optimal $L(\cdot)$)

Let $P = (\frac{1}{2}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10}, \frac{1}{10})$. Since

$$\begin{aligned} p' &= 0.1 + 0.1 = 0.2 & p'' &= 0.1 + 0.1 = 0.2 \\ p^{(3)} &= 0.2 + 0.1 = 0.3 & p^{(4)} &= 0.2 + 0.3 = 0.5 \\ p^{(5)} &= 0.5 + 0.5 = 1, \end{aligned}$$

then $L(\mathcal{C}) = 1 + 0.5 + 0.3 + 0.2 + 0.2 = 2.2 = 2 + \frac{1}{5}$, which we have already seen.

Lecture 3: A Measure of Information

Optimality of Huffman Codes
$L(\mathcal{C})$ of Huffman Codes
Huffman Codes are Optimal
Source Extension
Information
$l(s)$ measure
$l(s)$ is uniquely defined
Entropy
$H(S)$ measure
Properties of $H(S)$
Epilogue

First, a Lemma

Say w_1, w_2 are *siblings* if they are on the form $t0, t1$ for some $t \in T^*$. For instance $1011\underline{0}$ and $1011\underline{1}$ are siblings.

Lemma

Every source \mathcal{S} has an optimal binary code \mathcal{D} in which two of the longest code words are siblings.

Proof.

We saw last session that “every source has an optimal r -ary code for each $r \geq 2$ ”. Pick one, \mathcal{D} with $r = 2$ which minimises $\sigma(\mathcal{D}) = \sum_i l_i$.

Lecture 3: A Measure of Information

Optimality of Huffman Codes
 $L(\mathcal{C})$ of Huffman Codes
 Huffman Codes are Optimal
 Source Extension

Information
 $I(s)$ measure
 $I(s)$ is uniquely defined

Entropy
 $H(\mathcal{S})$ measure
 Properties of $H(\mathcal{S})$

Epilogue

First, a Lemma

Say w_1, w_2 are *siblings* if they are on the form $t0, t1$ for some $t \in T^*$. For instance $1011\underline{0}$ and $1011\underline{1}$ are siblings.

Lemma

Every source \mathcal{S} has an optimal binary code \mathcal{D} in which two of the longest code words are siblings.

Proof.

We saw last session that “every source has an optimal r -ary code for each $r \geq 2$ ”. Pick one, \mathcal{D} with $r = 2$ which minimises $\sigma(\mathcal{D}) = \sum_i l_i$.

Choose a longest $w \in \mathcal{D}$. We have $w = tt \in T^* = \mathbb{Z}_2$. Let $\bar{t} = 1 - t$, that is, $\bar{t} = 0$ or 1 as $t = 1$ or 0 . If $t\bar{t} \in \mathcal{D}$, we are done.

Assume $t\bar{t} \notin \mathcal{D}$. \mathcal{D} is optimal, thus instantaneous, thus prefix. Since $|tt|$ is maximal and $t\bar{t} \notin \mathcal{D}$, w must be the only code word w with t as prefix. But then we could replace w with t , obtaining \mathcal{D}' which retains the prefix property, has $L(\mathcal{D}') \leq L(\mathcal{D})$ (thus optimal) and $\sigma(\mathcal{D}') < \sigma(\mathcal{D})$, contradicting our choice of \mathcal{D} . Thus $t\bar{t} \in \mathcal{D}$ must hold. □

Now, the Theorem

Theorem

If \mathcal{C} is a binary Huffman code for S , then \mathcal{C} is an optimal code for S .

Proof.

Since “Huffman codes are instantaneous”, \mathcal{C} must be instantaneous. We must show that $L(\mathcal{C})$ is *minimal*.

Lecture 3: A Measure of Information

Optimality of Huffman Codes

 $L(\mathcal{C})$ of Huffman Codes

Huffman Codes are Optimal

Source Extension

Information

 $I(s)$ measure $I(s)$ is uniquely defined

Entropy

 $H(S)$ measureProperties of $H(S)$

Epilogue

Now, the Theorem

Theorem

If \mathcal{C} is a binary Huffman code for S , then \mathcal{C} is an optimal code for S .

Proof.

Since “Huffman codes are instantaneous”, \mathcal{C} must be instantaneous. We must show that $L(\mathcal{C})$ is *minimal*. Induction in q .

$q = 1$: then $\mathcal{C} = \{\epsilon\}$, thus $L(\mathcal{C}) = 0$ (\mathcal{C} is not a code, but we can make the best code from \mathcal{C}).

q , assuming theorem holds for $q - 1$: Huffman-reduce S one step, obtaining S' with $q - 1$ symbols $s_1, \dots, s_{q-2}, s' = s_{q-1} \vee s_q$. By $(\mathcal{C}, \mathcal{C}'\text{-Ldiff})$ we have $L(\mathcal{C}) - L(\mathcal{C}') = p_{q-1} + p_q = p'$.

Let $\mathcal{D} : s_i \mapsto x_i$ be an optimal binary code for S with a sibling pair of longest code-words $x_u = x0$, $x_v = x1$. Words attached to s_{q-1}, s_q must be longest, so assume wlg that $u = q - 1$ and $v = q$.

Form \mathcal{D}' for S' by $s_i \mapsto x_i$ for $1 \leq i \leq q - 2$, and $s' \mapsto x$. We have $L(\mathcal{D}) - L(\mathcal{D}') = p_{q-1} + p_q = p' = L(\mathcal{C}) - L(\mathcal{C}')$. Thus, $L(\mathcal{D}') - L(\mathcal{C}') = L(\mathcal{D}) - L(\mathcal{C})$. Since \mathcal{C}' is a Huffman code for S' , optimal by induction hypothesis, we must have $L(\mathcal{C}') \leq L(\mathcal{D}')$.

Thus $L(\mathcal{C}) \leq L(\mathcal{D})$. Since \mathcal{D} is optimal, so then must \mathcal{C} .

